

HTL Ybbs
Informationstechnologie
Ausbildungsschwerpunkt Netzwerktechnik



Diplomarbeit

Automatisierung von Installationsroutinen in Netzwerken

von

David Gnedt

geboren am 18. Oktober 1989 in Amstetten

und

Manuel Steiner

geboren am 6. Mai 1990 in Scheibbs

15. Mai 2009

Betreuer:

DI Andreas Brachinger

DI Christian Hammer

Präambel

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Unterschriften der Projektmitglieder

Ybbs, am 15.05.2009

David Gnedt	Manuel Steiner

Kurzfassung

Automatisierung von Installationsroutinen in Netzwerken

Diese Diplomarbeit beschäftigt sich mit automatischen Installationssystemen (Cloningsystemen), welche die Möglichkeit bieten, eine exakte Kopie des Inhaltes einer Festplatte eines Computers auf die eines anderen zu übertragen. Mit solchen Systemen kann der administrative Aufwand für die Einrichtung des Betriebssystems und der benötigten Software deutlich reduziert werden. Dies führt wiederum zu Kosteneinsparungen.

Es werden bereits bestehende Cloningsysteme verglichen und aufbauend darauf ein eigenes System entwickelt. Aufgrund der Tatsache, dass im Schulzentrum Ybbs a.d. Donau bis zum aktuellen Zeitpunkt (Schuljahr 2008/2009) eine kommerzielle Lösung im Einsatz ist und die verglichenen Systeme, welche frei erhältlich sind, nicht den wünschenswerten Leistungsumfang bieten können, wird ein eigenes System implementiert. Dieses soll entsprechend den Anforderungen im Schulzentrum Ybbs a.d. Donau entwickelt werden. Besonderes Augenmerk ist dabei auf einfache Bedienung und möglichst weitgehende Automatisierung des Systems zu legen. Durch die geplante Veröffentlichung als Open Source Projekt wird die Einsatzmöglichkeit jedoch nicht nur auf das Schulzentrum beschränkt.

Das Projekt gliedert sich in 3 Phasen, in welche auch diese Diplomarbeit grob gegliedert ist.

- In der Analysephase wird der Vergleich bereits existierender Lösungen durchgeführt.
- Der Anwendungsentwurf dient zur Findung geeigneter Methoden zur Softwareimplementierung.
- In der Anwendungsentwicklung werden die Anforderungen in ein konkretes Softwareprodukt umgesetzt.

Beim Anwendungsentwurf ist darauf zu achten, dass Software, welche von der eigens implementierten Lösung benutzt wird, unter einer freien Lizenz erhältlich ist. Somit besteht das entwickelte System ausschließlich aus freien Komponenten. Dieser Schritt ist notwendig, um eine Veröffentlichung als Open Source Projekt durchführen zu können.
Weiters wird in dieser Phase darauf geachtet, ein möglichst modulares System zu entwerfen, damit die einzelnen Softwareteile auf mehrere Computer verteilt werden können, Softwareteile einfach durch eine gewünschte andere Software ersetzt werden können und es einfach möglich ist, neue Komponenten zum System hinzufügen zu können.

Abstract

Automation of installation routines in networks

This diploma thesis deals with automatic installation systems (cloning systems) which offer the posibility to transfer an exact copy of the content of a computer's harddisk to other ones. With such systems, the administrative effort for the installation of an operating system and used programs can considerably be reduced; this aaves costs.

Already existing cloning systems are compared to each other. Based on this comparison, a new system is developed. A commercial product has been used in Schulzentrum Ybbs a.d. Donau until now (school year 2008/2009), but both this programme and the compared systems, which are freely available, can't provide the desired range of functions; for this reason an own system is implemented. This system should focus on easy usage and extensive automation. Due to the planned publication of the programme as open source, the usage is not restricted to the Schulzentrum.

The project is divided into 3 phases corresponding to the organization of the diploma thesis.

- In analysis phase the comparison of already existing cloning system is made.
- The application layout phase is used for finding suitable methods for the software implementation.
- In the application development phase the requirements are converted into a concrete software product.

In the application layout phase it has to be taken care of the fact that the software used by the self implemented solution is available under a free license. So the developed system only consists of free components, which is nessecary to be able to publish the software as an open source project.
Moreover in this phase it is to focus on implementing a modular system so that particular software parts can be spread over more than one computer, so that software parts can easily be replaced by other software and that new components can be added to the system.

Danksagung

Wir, David Gnedt und Manuel Steiner, bedanken uns bei unseren Betreuern DI Andreas Brachinger und DI Christian Hammer für die Unterstützung und Betreuung während des gesamten Projektverlaufes.

Bei DI (FH) Georg Ungerböck bedanken wir uns für die Unterstützung bei der Auswahl einer geeigneten Programmiersprache für die Entwicklung von Web Services. Dank seiner Informationen konnte die richtige Sprache gewählt werden, mit welcher das im selbst entwickelten System eingesetzte Web Service implementiert wurde.

Dem Schulerhalter, der Stadtgemeinde Ybbs a.d. Donau, danken wir für die zur Verfügung gestellte Infrastruktur, in der wir unser System im Schulzentrum installieren konnten und für die Benutzung der im Schulzentrum verfügbaren Hardware, mit der es uns möglich war, den Vergleich bereits existierender Cloningsysteme sowie das Testen des selbst entwickelten Systems durchzuführen.

Der Polytechnischen Schule St. Peter in der Au, sowie dem dortigen IT-Administrator Ing. Thomas Gnedt, danken wir für die zur Verfügung gestellte Infrastruktur zum Testen des Cloningsystems.

Weiters danken wir dem Inhaber der Firma Management Technik Systeme, Martin Steiner, für den Druck der Diplomarbeit.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Dieses Dokument beinhaltet den Vergleich einiger bereits existierender Cloninglösungen und die Beschreibung der Entwicklung eines eigenen Systems, welches auf die gewonnenen Informationen des Vergleichs aufbaut und zusätzliche Funktionen, für den Einsatz am Schulzentrum implementiert.

Im Kapitel 2 (Analyse) werden die generellen Anforderungen an ein Cloningsystem festgehalten. Danach wird der Vergleich von 5 existierenden Systemen erläutert. Dabei werden der Funktionsumfang der einzelnen Produkte sowie erzielte Zeit- und Geschwindigkeitswerte in einem praktischen Test berücksichtigt. Weiters werden einige wichtige Merkmale jedes der verglichenen Systeme aufgezeigt. Durch die Informationen aus den Tests werden die Nachteile dieser Lösungen zusammengefasst und die Motivation, ein eigenes System zu implementieren, beschrieben.

Im folgenden Kapitel 3 (Lösungsansatz) wird erläutert, mit welchen Hilfsmitteln und Methoden die Implementierung der eigenen Lösung erfolgt. Dabei wird das gesamte entworfene System beschrieben und veranschaulicht. Weiters werden wichtige Teilkomponenten und deren Implementierungsansatz festgehalten. Im Anschluss werden zur Verfügung stehende Programmiersprachen verglichen und geeignete Sprachen für die Entwicklung der Einzelteile der Gesamtlösung ausgewählt.

Das Kapitel 4 (Anwendungsentwicklung) widmet sich der Implementierung der neu entworfenen Lösung. Dabei wird näher auf die verwendeten Techniken eingegangen, welche zur Entwicklung beitragen. Es werden die aufgetretenen Schwierigkeiten und die Vorgehensweise der Bewältigung dieser beschrieben. Zur praktischen Veranschaulichung der Entwicklung dienen diverse praktische Sourcecodebeispiele. Dieses Kapitel soll den Einsatz und die grundlegende Funktionsweise der verwendeten Techniken herausheben.

Im abschließenden Kapitel 5 (Ergebnisse) ist eine Auswertung des entwickelten Systems aus der Sicht des Projektmanagements enthalten. Des Weiteren wird die Veröffentlichung der Software als Open Source Projekt erläutert und in einer Roadmap die weiteren Tätigkeiten bei der Entwicklung festgehalten. Außerdem wird die Installation und der Einsatz der entwickelten Lösung im Schulzentrum Ybbs a.d. Donau beschrieben.

Kapitel 2

Analyse

Dieses Kapitel beschreibt den aktuellen Vorgang, um die Computer im Schulzentrum für den produktiven Einsatz zu präparieren und widmet sich dem Vergleich bereits bestehender Cloningsysteme. Dieser Vergleich beinhaltet die unterschiedlichen Funktionen der getesteten Produkte sowie die Analyse eines kompletten Sicherungs- und Wiederherstellungsvorganges einer Festplatte in Bezug auf gemessene Zeiten. Weiters werden die Gründe für ein eigenes System erläutert.

In weiterer Folge wird von Server- und Client-Computer ausgegangen. Der Server stellt das System dar, auf dem die Festplattenabbilder gespeichert werden und die Steuerung der Cloningsoftware durch den Benutzer vorgenommen wird. Der Client hingegen ist ein PC, von dem ein Image der Festplatte angefertigt oder ein bereits existierendes auf dessen Platte übertragen wird.

2.1 Aktuelle Situation im Schulzentrum

Die Computerräume im Schulzentrum sind grundsätzlich so ausgestattet, dass in jedem Raum die selbe Hardware für jeden Computer vorhanden ist. Als Betriebssystem wird zum Großteil Windows XP eingesetzt. Es gibt jedoch auch Räume mit Windows XP und SuSE Linux als Dualbootsystem, wobei GRUB als Bootmanager dient.

Aktuell wird zur Imageerzeugung und zur Übertragung eines solchen auf einen oder mehrere Computer ein kommerzielle Produkt verwendet. Dabei wird die Steuerung über einen zentralen Multicast-Server vorgenommen, der üblicherweise auf dem Rechner im Raum ausgeführt wird, auf dem der Lehrer arbeitet. Es ist also erforderlich, in jedem Raum diesen Server zu betreiben. Die Images für die Clients werden auf einer portablen Festplatte gespeichert, um sie in jedem Raum einsetzen zu können.

2.1.1 Cloningvorgang

Um die Clients für den Cloningvorgang vorzubereiten, wird auf ihnen das Programm sysprep ausgeführt. Dieses dient nach dem Vorgang zur Integration des Clients in die Schuldomäne. Die Computer werden aus der Windows Domäne herausgenommen. Weiters wird von AVG ein Tool zur Vorbereitung des Clonens der Antivirus Programmlizenz ausgeführt. Nach dieser Vorbereitung werden die Compu-

ter, von denen ein Festplattenimage erstellt, oder ein Abbild auf die Festplatte übertragen werden soll, über PXE gebootet. Dabei wird ein eigens angepasstes PXE-Image vom zentralen DHCP/TFTP Server übertragen. Auf dem Computer mit installiertem Multicast-Server wird dann der Vorgang zur Imageerstellung beziehungsweise Rücksicherung gestartet, sobald sich alle Clients über PXE gemeldet und das Ghost Programm gestartet haben. Die Rücksicherung auf mehrere Computer wird als Multicast über das Netzwerk übertragen. Nach erfolgreich ausgeführtem Cloningvorgang werden die Computer neu gestartet. Dabei wird automatisch das zuvor vorbereitete Programm sysprep ausgeführt und übernimmt nach Eingabe des Rechnernamen, des lokalen Administratorpasswortes und der Administrator-Zugangsdaten zur Domäne auf jedem Client die Integration in die Schuldomäne.

Falls ein Dualbootsystem mit Windows XP und SuSE Linux auf die Festplatten der Clients übertragen wird, muss danach der Master-Boot-Record manuell neu geschrieben werden, weil dieser unmittelbar nach dem Cloningvorgang nicht richtig funktioniert.

2.2 Anforderungen an ein Cloning System

Aus dem bereits bestehenden Prozess ergeben sich folgende essentielle Anforderungen an ein Cloningsystem, welches im Schulzentrum eingesetzt werden soll.

- Einfache Bedienung des Systems
  Das System soll ohne größere Einarbeitungszeit bedient werden können und weitestgehend automatisch agieren. Durch den Einsatz eines Webinterface als Steuerungssoftware für den Benutzer und Verwendung von Wizards zum Anlegen und Rücksichern der Images kann der Bedienkomfort erhöht werden.

- Verwaltung von Hosts
  Clients sollen in Gruppen unterteilt werden können, um ein Image ohne größeren Aufwand auf eine ganze Gruppe von Computer gleichzeitig übertragen zu können. Außerdem wird dadurch die Übersichtlichkeit in der Benutzersoftware erhöht. Weiters wäre die Möglichkeit, Detailinformationen, wie die Seriennummer eines Hosts, speichern zu können, von Vorteil.

- Imageübertragung auf mehrere Computer gleichzeitig
  Ein vorhandenes Image soll auf mehrere Clients synchron übertragen werden. Dies wird mit dem Einsatz von Multicast-Übertragung realisiert. Das Image wird dabei nur einmal in das Netzwerk gesendet. Im Gegensatz zu einem Broadcast, wo alle Computer des selben Subnetzes das Image empfangen würden, erhalten es bei Multicast nur jene Computer, welche es benötigen.

2.3 Vergleich bereits existierender Systeme

Dieser Vergleich dient dem Aufzeigen von Stärken und Schwächen von bestehenden Cloningsystemen. Weiters beinhaltet er eine individuelle Stellungnahme zu jedem Produkt mit Informationen zum praktischen Imagingvorgang und eventuell dabei aufgetretene Schwierigkeiten.

2.3.1 Produktübersicht

Folgende Produkte wurden verglichen:

- Acronis SnapDeploy 3 für Workstation
- DRBL 1.9.1-26 (Clonezilla Server Edition)
- FOG 0.23
- Norton Ghost 2005
- OpenClone, die im Rahmen der Diplomarbeit angefertigte Cloninglösung

Bei den beiden kostenpflichtigen Produkten von Acronis und Norton wurden Testlizenzen verwendet, um den praktischen Test durchführen zu können.

2.3.2 Testsystem

Zum Messen der Übertragungsdauer eines Abbildes wurden zwei Rechner direkt mittels einem Crossover-Netzwerkkabel verbunden, sodass eine Gigabit-Verbindung hergestellt war. Bei einer Verwendung eines Switches zwischen den beiden Testcomputern muss darauf geachtet werden, dass entweder das Spanning Tree Protocol zur Gänze deaktiviert oder die Initialisierungsphase zeitlich so beschränkt wird[1], dass ein PXE Client eine erfolgreiche DHCP Anfrage stellen kann. Dazu muss diese Spanning Tree Phase für den Port, an welchen der Computer angeschlossen ist, abgeschlossen sein, bevor der Client das Timeout für die DHCP Anfrage erreicht. Der Computer wiederholt die DHCP Anfrage mehrmals, dadurch wird jedoch auch die Netzwerkverbindung neu initialisiert. Dies führt wiederum zu einer erneuten Spanning Tree Initialisierungsphase.

Im Folgenden werden die beiden Testrechner Server und Client beschrieben. In beiden Computern war bis auf die Festplatte die gleiche Hardware verbaut. Auf der zu sichernden Festplatte war ein Dualbootsystem mit den Betriebssystemen Windows XP mit Service Pack 2 und SuSE Linux 10.3 installiert. Als Bootmanager diente GRUB.

Gemeinsame Hardware

- Name des Komplettsystems: Fujitsu Siemens P5925
- Prozessor: Intel Core2 Duo E6550 (Taktfrequenz: 2,33 Gigahertz)
- RAM: 2 Gigabyte
- Netzwerkkarte: 1 Gigabit Intel PRO/1000 GT Desktop Adapter

[1] z.B. durch den Einsatz von Rapid Spanning Tree

Server

- Festplatte: Samsung Spinpoint T166 HD501LJ (500 Gigabyte Kapazität, 16 Megabyte Cache, SATA2 Anschluss)

Client

- Festplatte: Seagate Barracuda 7200.10 ST3250310AS (250 Gigabyte Kapazität, 8 Megabyte Cache, SATA2 Anschluss)
  - Primäre Partition 1: 39998 Megabyte Gesamtgröße (11974 Megabyte belegt), Dateisystem: NTFS v3.1, Inhalt: Windows XP
  - Primäre Partition 2: 149998 Megabyte Gesamtgröße (53562 Megabyte belegt), Dateisystem: NTFS v3.1, Inhalt: Daten
  - Logische Partition 1: 2055 Megabyte Gesamtgröße, Dateisystem: Linux SWAP
  - Logische Partition 2: 46422 Megabyte Gesamtgröße (6927 Megabyte belegt), Dateisystem: ext3, Inhalt: SuSE Linux

2.3.3 Vergleichstabelle

In der Tabelle 2.1 ist die Gegenüberstellung der getesten Produkte veranschaulicht. Sie umfasst sowohl den Funktionsumfang als auch Details zum praktischen Cloningtest.

2.3.4 Acronis SnapDeploy 3 für Workstation

Bei diesem Produkt gestaltete sich bereits das Booten des Clients über PXE schwierig. Es war zwar ein PXE-Server im Produkt enthalten, jedoch kein DHCP-Server, der dem Client eine IP-Adresse zuweisen konnte. Das Benutzerhandbuch schlägt einen Windows DHCP-Server vor, welcher aber nur in Windows Server Versionen enthalten ist. [Inc09b] Es wurde also ein DHCP-Server mit konfigurierbarem 'nextserver' Attribut benötigt. Zum Einsatz kam der frei erhältliche dhcpsrv in der Version 1.6.4[2]. Mit diesem Server war es möglich, auf den in Acronis enthaltenen TFTP-Server weiter zu verweisen, um die erforderlichen Daten zum Client übertragen zu können. Da der TFTP-Server unter Windows jedoch als Service registriert war und auch die Ports für DHCP (67 und 68) belegte, musste der DHCP Server vor dem Acronis TFTP Server gestartet werden. Erst danach konnte der Client ordnungsgemäß booten und das Acronis SnapDeploy Programm ausführen. Nach erfolgreichem Anlegen eines Festplattenabbildes war das Zurückspielen jedoch nicht mehr möglich. Laut Acronis war trotz vorhandener Testlizenz keine Serverlizenz vorhanden. Somit weigerte sich das SnapDeploy, das Image wieder auf den Client zu übertragen.

Die Software benötigte die zweitlängste Zeit für das Erstellen des Festplattenabbildes. Dieses Image benötigte unter allen Images den meisten Speicherplatz. Somit sprechen diese Werte gegen das Programm.

[2] Frei verfügbar unter `http://ruttkamp.gmxhome.de/dhcpsrv/dhcpsrv.htm`

2.3.5 DRBL 1.9.1-26 (Clonezilla Server Edition)

DRBL steht für 'Diskless Remote Boot in Linux'. Es wurde im Clonzilla Box Mode (SSI: Singe System Image) betrieben. Dabei sind am Server mindestens 2 Netzwerkinterfaces notwendig. Eines ist für den Zugang des Servers zum Internet zuständig und das andere für die Kommunikation mit den Clients. Falls im Serversystem nicht zwei Netzwerkkarten verfügbar sind, lässt sich diese Software nicht ordnungsgemäß ausführen. Die Clonezilla Server Edition wurde unter Ubuntu Linux installiert und betrieben. Da diese Cloningsoftware bereits alle benötigten Serverdienste enhielt, konnte der Client sofort über PXE gestartet werden und ein Image seiner Festplatte angelegt werden. Zur Komprimierung wurde gzip ausgewählt. Durch die Textausgaben von Partimage und ntfsclone konnte jederzeit die Übertragungsgeschwindigkeit des unkomprimierten Festplatteninhaltes und der Fortschritt am Client festgestellt werden. Es ergaben sich folgende durchschnittliche Übertragungsgeschwindigkeiten.

- Primäre Partition 1: 1236 MiB pro Minute (mit ntfsclone)
- Primäre Partition 2: 1320 MiB pro Minute (mit ntfsclone)
- Logische Partition 2: 1460 MiB pro Minute (mit Partimage)

Clonezilla Server Edition arbeitete mit einer Imagingzeit von 48 Minuten 11 Sekunden und einer Zeit zum Rücksichern des Images von 29 Minuten 15 Sekunden am schnellsten unter den gestesteten Produkten. Auch die Größe des Festplattenabbildes als zweitkleinste konnte überzeugen.

2.3.6 FOG 0.23

FOG wurde ebenfalls unter Ubuntu Linux verwendet. Es bot als einziges getestetes Produkt ein Webinterface zur Konfiguration. Dies bietet natürlich einen Vorteil für die Administation, weil die Steuerung der Software nicht auf einen Computer mit installierter Software beschränkt ist. Es waren ebenfalls bereits alle benötigten Serverdienste im Softwarepaket enthalten. Die Komprimierung des erstellten Images wurde mit gzip durchgeführt. Es bestanden hier keine weiteren Alternativen zur Auswahl. FOG erreichte folgende durchschnittliche Übertragungsgeschwindigkeiten.

- Primäre Partition 1: 117 MiB pro Minute (mit Partimage)
- Primäre Partition 2: 116 MiB pro Minute (mit Partimage)
- Logische Partition 2: 96 MiB pro Minute (mit Partimage)

Diese Geschwindigkeiten und die daraus resultierende Imagingdauer von mehr als 10 Stunden waren mit Abstand die schlechtesten Werte im Vergleich. Auch die Rücksicherung dauerte mit über 8 Stunden inakzeptabel lange. Trotz des Webinterface und der geringsten Imagegröße von 42676 MiB disqualifizierte sich FOG mit diesen langen Durchführungszeiten für den sinnvollen Einsatz als Cloningsoftware. Erstaunlich dabei war, dass die mit ext3 formatierte logische Partition noch langsamer als die beiden NTFS Partitionen gesichert wurden, obwohl Partimage eigentlich wesentlich besser mit ext3 als NTFS zurechtkommen sollte, weil NTFS von Partimage nur experimentiell unterstützt wird.

2.3.7 Norton Ghost 2005

Damit Norton Ghost auf den Clients mittels PXE verwendet werden konnte, mussten zuerst ein DHCP- und TFTP-Server für Windows am Server installiert werden, denn Norton Ghost enthält keinen der beiden Dienste. Es wurde wie bereits bei SnapDeploy dhcpsrv in der Version 1.6.4 als DHCP-Server verwendet. Für den TFTP-Server kam tftpd32 in Version 3.28 zum Einsatz[3]. Nach erfolgreicher Konfiguration der Dienste konnte der Client aufgrund eines zu großen PXE Images, welches am Client nicht gespeichert werden konnte, nicht booten. Der Fehlercode lautete: PXE-E79: NBP is too big to fit in free base memory. Dieses PXE-Image wurde mit Norton Ghost erstellt. Damit der Client mittels PXE booten konnte, wurde ein PXE-Image mithilfe der Universal TCP/IP Network Bootdisk erstellt[4]. Der Client startete erfolgreich und der praktische Test konnte durchgeführt werden.

Norton Ghost erreichte sowohl beim Erstellen als auch beim Rücksichern die zweitbeste Ausführungszeit. Bei der Größe des Festplattenabbildes reihte sich Ghost allerdings ganz hinten an. Dies und der höhere Konfigurationsaufwand im Gegensatz zu anderen getesteten Produkten stellen die gute Imagingdauer etwas in den Schatten.

2.4 Fazit

Die getesteten Systeme haben alle an verschiedenen Stellen entscheidende Schwächen, so dass sie sich nicht optimal für den Einsatz, vor allem im Schulzentrum, eignen. Für den Benutzer bietet nur FOG eine grafische Oberfläche, welche, unabhängig von einem bestimmten Computer, aus dem gesamten Netzwerk mittels eines Webbrowser erreichbar ist. Clonezilla Server Edition kann zwar auch über das Netzwerk erreicht werden, falls auf dem Server ein Secure Shell Zugang aktiviert ist, jedoch bietet dieses Programm keine vergleichbar benutzerfreundliche Oberfläche wie FOG. Die Produkte von Acronis und Norton sind zwar als grafisch zu bedienende Programme am Server vorhanden, jedoch müsste der Benutzer zusätzlich Windows Remote Desktop oder ein anderes Produkt wie zum Beispiel VNC verwenden, um vom Netzwerk auf die Software zugreifen zu können. Dies ist umständlicher als die Verwendung eines Webinterface. FOG ist allerdings, abgesehen vom Webinterface und der Imagegröße, aufgrund der inakzeptablen Dauer für das Erstellen und Rücksichern eines Abbildes, für den Einsatz ungeeignet. Die einfache Verwaltung von Hosts und Gruppen ist ebenfalls nur in FOG möglich. Acronis SnapDeploy, Clonezilla Server Edition sowie Norton Ghost bieten hier keine Möglichkeiten. Ein weiterer Nachteil von SnapDeploy und Ghost sind die anfallenden Lizenzkosten.
Der Vergleich und die daraus gewonnenen Informationen waren Motivation zur Entwicklung einer eigenen Cloningsoftware. Sie soll so entwickelt werden, dass sie besser im Schulzentrum eingesetzt werden kann, als das aktuell verwendete Produkt. Dabei ist auf einfache Bedienbarkeit und weitestgehende Automatisierung der Imaging- und Rücksicherungsprozesse Wert zu legen. Das System befindet sich ebenfalls in der Vergleichstabelle (OpenClone 0.1.0, siehe Tabelle 2.1)

[3]Frei verfügbar unter `http://tftpd32.jounin.net/`
[4]Frei verfügbar unter `http://netbootdisk.com`

	Acronis SnapDeploy 3 für Workstation	**DRBL 1.9.1-26 (Clonezilla Server Edition)**
Betriebssystemunterstützung		
Server	Windows[a]	Linux[b]
Client	unabhängig	unabhängig
Benutzeroberfläche		
Server (Text/Grafisch/Web)	Grafisch	Text
Client (Text/Grafisch/Web)	Grafisch	Text
Client Bootmethoden		
CD-ROM	*	*
Netzwerk (PXE/Wake on LAN)	*/*	*/*
geforderte Benutzerinteraktionen		
am Server (Abbilderstellung/-rücksicherung)	keine/notwendig	notwendig/notwendig
am Client (Abbilderstellung/-rücksicherung)	notwendig/keine	möglich/möglich (am Server auswählbar)
Abbildumfang		
gesamte Festplatte (alle Partitionen+MBR[c])	*	*
einzelne Partition	*	*
Komprimierung	Normal, Hoch, Maximum	gzip, bzip2, lzo
unterstützte Dateisysteme		
FAT32 (intelligent/Sektor)	*/*	*/* (Partimage/dd)
NTFS (intelligent/Sektor)	*/*	*/* (ntfsclone/dd)
ext3 (intelligent/Sektor)	*/*	*/* (Partimage/dd)
Linux SWAP (intelligent)	*	*
Imageübertragung		
Unicast (Erstellen/Rücksichern)		
SMB	*/-	-/-
FTP	*/-	-/-
NFS	-/-	*/*
eigenes Protokoll	-/*	-/-
Rücksicherung mit Multicast	*	*
Nachbearbeitungsmethoden		
Hostname setzen (Windows/Linux)	*/-	*/- (DRBL-winroll/-)
statische IP setzen (Windows/Linux)	*/-	-/-
neue SID generieren (Windows)	*	* (DRBL-winroll)
Domainintegration (Windows/Linux)	*/-	-/-
Hostmanagement (Hosts/Gruppen)	-/-	*/-
Tests		
Imageerstellung (Dauer, Imagegröße)	57 Min 9 Sek, 46113 MiB	48 Min 11 Sek, 43989 MiB
Imagerücksicherung (Dauer)	nicht möglich[d]	29 Min 15 Sek
Lizenzkosten	22,55 Euro pro Rechner [Inc09a]	- (Open Source)

[a] Windows 2000, XP, Vista; Server nur mit Server Lizenz
[b] Debian, Ubuntu, SuSE, RedHat, RHEL, Fedora, Mandriva, CentOS, Scientific Linux, BSD
[c] Master Boot Record
[d] fehlende Serverlizenz

	FOG 0.23	**Norton Ghost 2005**
Betriebssystemunterstützung		
Server	Linux[a]	Windows
Client	Windows[b]	unabhängig
Benutzeroberfläche		
Server	Web	Grafisch
Client	Text	Grafisch
Client Bootmethoden		
CD-ROM	-	*
Netzwerk (PXE/Wake on LAN)	*/*	*[c]/-
geforderte Benutzerinteraktionen		
am Server (Abbilderstellung/-rücksicherung)	notwendig/notwendig	notwendig/notwendig
am Client (Abbilderstellung/-rücksicherung)	keine/keine	keine/keine
Abbildumfang		
gesamte Festplatte (alle Partitionen+MBR)	*	*
einzelne Partition	*	*
Komprimierung	gzip	Normal, Hoch
unterstützte Dateisysteme		
FAT32 (intelligent/Sektor)	*/* (Partimage/dd)	*/*
NTFS (intelligent/Sektor)	*/* (Partimage/dd)	*/*
ext3 (intelligent/Sektor)	-/-	*/*
Linux SWAP (intelligent)	-	*
Imageübertragung		
Unicast (Erstellen/Rücksichern)		
SMB	-/-	-/-
FTP	-/-	-/-
NFS	*/*	-/-
eigenes Protokoll	-/-	*/*
Rücksicherung mit Multicast	*	*
Nachbearbeitungsmethoden		
Hostname setzen (Windows/Linux)	*/-[d]	-/-
statische IP setzen (Windows/Linux)	-/-	-/-
neue SID generieren (Windows)	-	-
Domainintegration (Windows/Linux)	*/-[d]	-/-
Hostmanagement (Hosts/Gruppen)	*/*	-/-
Tests		
Imageerstellung (Dauer, Imagegröße)	10 Std 35 Min, 42676 MiB	49 Min 53 Sek, 48931 MiB
Imagerücksicherung (Dauer)	8 Std 48 Min	38 Min 50 Sek
Lizenzkosten	- (Open Source)	19,50 Euro pro Rechner [Gmb09]

[a]Debian, Ubuntu, RedHat, Fedora, CentOS

[b]Windows 2000, XP, 2003, Vista

[c]Nur mit zusätzlicher Software wie der Universal TCP/IP Network Bootdisk (`http://www.netbootdisk.com/`) und geeignetem DHCP- und TFTP-Server möglich

[d]mit FOG Client Service

	OpenClone 0.1.0
Betriebssystemunterstützung	
Server	Linux (Ubuntu)
Client	unabhängig
Benutzeroberfläche	
Server	Web
Client	Text
Client Bootmethoden	
CD-ROM	-
Netzwerk (PXE/Wake on LAN)	*/-
geforderte Benutzerinteraktionen	
am Server (Abbilderstellung/-rücksicherung)	notwendig/notwendig
am Client (Abbilderstellung/-rücksicherung)	keine/keine
Abbildumfang	
gesamte Festplatte (alle Partitionen+MBR)	*
einzelne Partition	-
Komprimierung	gzip, lzo
unterstützte Dateisysteme	
FAT32 (intelligent/Sektor)	*/- (Partimage/-)
NTFS (intelligent/Sektor)	*/- (Partimage/-)
ext3 (intelligent/Sektor)	*/- (Partimage/-)
Linux SWAP (intelligent)	*
Imageübertragung	
Unicast (Erstellen/Rücksichern)	
SMB	-/-
FTP	-/-
NFS	-/-
eigenes Protokoll	*/*
Rücksicherung mit Multicast	*
Nachbearbeitungsmethoden	
Hostname setzen (Windows/Linux)	-/-
statische IP setzen (Windows/Linux)	-/-
neue SID generieren (Windows)	-
Domainintegration (Windows/Linux)	-/-
Hostmanagement (Hosts/Gruppen)	*/*
Tests	
Imageerstellung (Dauer, Imagegröße)	34 Min 23 Sek, 47358 MiB
Imagerücksicherung (Dauer)	49 Min 4 Sek
Lizenzkosten	- (Open Source)

Tabelle 2.1: Vergleich der Cloningsysteme

Kapitel 3

Lösungsansatz

In diesem Kapitel wird beschrieben, welche Überlegungen zur Realisierung eines eigenen Cloningsystems gemacht und welche Entscheidungen zum Aufbau getroffen wurden.

3.1 Systemarchitektur

Durch die Anforderungen wird bereits festgelegt, dass es sich um ein Client-Server System handeln muss, welches die Images auf einem zentralen Server speichert, von dem aus diese auf alle Clients verteilt werden können. Weiters ist es für eine einfache Handhabung notwendig alle Informationen an einer zentralen Stelle zur Verfügung zu haben. Somit ergibt sich ein System mit drei Komponenten.

- Server, der alle Images speichert, den Cloningprozess koordiniert und die Benutzeraktionen verarbeitet
- Client, der die Aufgaben vom Server erhält und den Cloningvorgang ausführt
- Benutzeroberfläche mit welcher der Server gesteuert werden kann

3.2 Clientsoftware

Da das Cloningsystem vom eingesetzten Betriebssystem am Client unabhängig sein soll, muss auf dem Client ein eigenes Betriebssystem für die Clientsoftware gestartet werden. Eine betriebssystemunabhängige Clientsoftware, welche beispielsweise in Java geschrieben sein könnte, kommt nicht in Frage, da für einen erfolgreichen Cloningvorgang die Festplatte im Client in einem konsistenten Zustand sein muss, was überlicherweise bei einem laufenden Betriebssystem nicht garantiert werden kann. Das für die Clientsoftware gestartete Betriebssystem muss also unabhängig von der Festplatte im Clientrechner sein. Dazu gibt es die Möglichkeit ein Betriebssystem von einem Wechseldatenträger, also beispielsweise einer Diskette, CD, DVD oder einem USB-Stick, zu starten. Dazu wäre es aber notwendig, für einen Cloningvorgang jeden Client mit einem Wechseldatenträger zu starten, was schon bei einer geringen Anzahl an Rechnern einen erheblichen Aufwand bedeutet. Daher bleibt nur die Option, den Client

über Netzwerk zu booten, was etwas mehr Einrichtungsaufwand am Server bedeutet, jedoch das Starten der Clients mit einem eigenen Betriebssystem sehr vereinfacht. Die bei heutigen PCs eingesetzte Netzwerk-Boot Methode heißt Preboot eXecution Environment.

3.3 Prozessorganisation

Im Vergleich von existierenden Systemen wurde festgestellt, dass nur FOG als einziges Cloningsystem der Anforderung zur Verwaltung von Hosts genügt und alle anderen Lösungen entweder gar keine oder nur eine geringfügige Möglichkeit zur Verwaltung bieten. Durch den guten Ansatz von FOG zur Verwaltung von Hosts wurde dieses näher analysiert, um der gegebenen Einschränkung auf Windows Betriebssysteme auf den Grund zu gehen. FOG implementiert mehrere Modi zum Erstellen eines Abbildes, dabei gibt es einen Modus für einzelne NTFS Partitionen, welcher eine Veränderung der Partitionsgröße beim Wiederherstellen erlaubt, einen für mehrere Partitionen auf einer Festplatte, einen für mehrere Festplatten mit mehreren Partitionen und einen für ein komplettes Byte-für-Byte Abbild mit dd. Zusätzlich wurde auch der Quellcode von FOG analysiert und herausgefunden, dass alle diese Modi sehr ineffizient programmiert worden sind. Obwohl jeder Modus sehr ähnliche Aktionen durchführt, wurde jeder einzelne komplett neu implementiert, auch die Sicherung des MBR. FOG unterstützt hauptsächlich deshalb kein Linux, da immer, abgesehen vom zeitraubenden Byte-für-Byte Abbild, die Imagingsoftware Partimage verwendet wird und Partimage leider keine Linux Swap Partitionen unterstützt. Genau diese unflexible Programmierung ist der große Nachteil von FOG und sollte im OpenClone Projekt nicht wiederholt werden. Daher wurde ein sehr modularer und flexibler Aufbau des OpenClone Projekts gewählt.

Der Grundgedanke war, die Sicherungs- oder Wiederherstellungsprozesse in mehrere Operationen zu untergliedern und für den Benutzer wieder in Tasks zusammenzufassen. Das bedeutet ein Client muss nichts vom eigentlichen Sicherungs- oder Wiederherstellungsprozess wissen, er führt einfach eine Operation nach der anderen aus. Wenn die Client/Server Infrastruktur einmal vorhanden ist, können somit jederzeit neue Operationen implementiert werden. Weiters sollen die vom Client ausgeführten Operationen möglichst gut gesteuert werden können, damit nicht eine eigentlich gleichwertige Operation mehrmals implementiert werden muss. So wurden die für einen einfachen Sicherungs- und Wiederherstellungsprozess notwendigen Grund-Operationen definiert.

- Warten (Idle)
- Partitionierung (Partition)
- Abbildung (Image)
- Herunterfahren (Shutdown)

Eine Vorraussetzung ist, dass beispielsweise die Abbildungsoperation mit unterschiedlicher Imagingsoftware arbeiten soll, damit mehrere Dateisysteme unterstützt werden können. Weiters wurde festgelegt, dass eine Abbildungsoperation nur das Abbild einer einzigen Partition erstellen kann. Es muss also bei einem Task zum Sichern einer Festplatte für jede Partition eine eigene Abbildoperation am Client

ausgeführt werden. Für den Benutzer erscheint der Sicherungsprozess allerdings trotzdem als einziger Task, um keine Verwirrung zu schaffen.

3.4 Client-Server Kommunikation

Die Kommunikation zwischen Client- und Server soll mit nur fünf Befehlen erfolgen. Dazu gehört das Registrieren eines Rechners, der Login des Clients, das Abrufen der nächsten Operation, das Aktualisieren des Status einer Operation und das Ausloggen des Clients. Mehr Befehle sollten auch für spätere Weiterentwicklung des Projekts nicht notwendig sein, da zur Erweiterung der Funktionalität, wie bereits beschrieben, nur neue Operationen hinzugefügt werden müssen, welche dann über den Befehl zum Abrufen der nächsten Operation und der Statusaktualisierung eingesetzt werden können.

Um den Zeitaufwand für die Implementierung der Kommunikation zwischen Client und Server möglichst gering zu halten, ist eine Technik wie RPC[1] zur Abstraktion von Vorteil, da der für eine Socket Kommunikation notwendige Protokollentwicklungs- und Programmieraufwand entfällt. Somit müssen nur die Funktionen am Server und die Funktionsaufrufe am Client programmiert werden und die Kommunikation über das Netzwerk erfolgt quasi automatisch. Für eine weitere Vereinfachung sollten als Parameter für Funktionsaufrufe auch komplexe Datentypen bzw. Objekte eingesetzt werden können. Eine relativ neue Technik, welche diese Möglichkeit bietet, sind Webservices. SOAP ist eine programmiersprachen- und plattformunabhängige Webserviceimplementierung, welche mittels XML Nachrichten üblicherweise über das HTTP Protokoll, jedoch auch über andere Protokolle kommunizieren kann. Bei SOAP werden die Methoden, sowie die Datentypen des Webservices mit Hilfe von WSDL[2] definiert, somit ist eine genaue Beschreibung des Webservices möglich. In nahezu jeder Programmiersprache gibt es eine SOAP Implementierung, welche eine sehr einfache Kommunikation über Proxyobjekte ermöglicht. Dabei wird am Client ein Objekt angelegt, welches alle Methoden des Webservices zur Verfügung stellt.

3.5 Benutzeroberfläche

Eine zentrale Anforderung ist die einfache Bedienbarkeit des Cloningsystems, dabei spielt die Schnittstelle zum Benutzer natürlich die größte Rolle. Die Benutzeroberfläche soll außerdem einfach zu installieren sein oder bestenfalls gar keine Installation benötigen. Wie im Vergleich der bestehenden Systeme erkannt werden konnte, ist hierfür ein Webinterface sehr von Vorteil, da es von jedem Rechner aus mit einem Browser ohne schwierige Abhängigkeiten oder lästigen Installationsaufwand verwendet werden kann. Außerdem ist das Webinterface plattformunabhängig und könnte z.B. sogar mit einem Mobiltelefon über WLAN aufgerufen werden. Ein weiterer Vorteil ist, dass ein Software Update nur am Server durchgeführt werden muss und automatisch die neue Version für alle Clients verfügbar ist. Bei einer eigenständigen Software zur Verwaltung müsste diese im Zuge eines Updates am Server ebenfalls auf allen Clients aktualisiert werden. Weiters soll die Benutzeroberfläche das Webinterfaces durch den Einsatz von Assistenten zum Erstellen und Wiederherstellen von Images möglichst einfach gehalten

[1] Remote Procedure Call
[2] Web Service Description Language

werden. Die etwas geringere Interaktivität eines üblichen Webinterfaces könnte durch den Einsatz von AJAX[3] sehr gut ausgeglichen werden. Ein Einsatz von AJAX ist jedoch nicht Teil dieser Diplomarbeit, sondern, wie in Kapitel 5.3.3 beschrieben, für die Zukunft geplant.

Die Erfahrungen mit Clonezilla Live, welches beim Starten in chinesischer Sprache nachfragt, welche Sprache für die Ausgaben benutzt werden soll, haben gezeigt, dass eine weit verbreitete Sprache als Standardsprache sinnvoll ist. Somit fiel die Entscheidung, Englisch als Standardsprache für das Webinterface zu verwenden, sehr einfach, da das Projekt zudem als OpenSource Projekt veröffentlicht werden soll und für einen möglichst großen Personenkreis einsetzbar sein soll.

3.6 Hostmanagement

Wie bereits erwähnt ist das Hostmanagement beim Cloningsystem FOG sehr gut gelöst, daher soll das eigene Cloningsystem einen ähnlichen Ansatz verfolgen. Dabei sollen von jedem Rechner Informationen gespeichert werden. Diese Informationen unterteilen sich in jene, die für die Funktion des Cloningsystems notwendig sind, wie beispielsweise die verfügbaren Festplatten des Rechners, sowie den Partitionen darauf und jenen Informationen, die nur den Benutzer unterstützen sollen, wie der eingebaute Prozessor, Arbeitsspeicher und die Seriennummer des Rechners. Die Hosts sollen dabei nicht händisch vom Benutzer erfasst werden müssen, sondern möglichst automatisch in das System integriert werden. Dazu ist es notwendig alle benötigten Informationen automatisch am Client zu erheben und bei der Registrierung, sowie beim Login zum Server zu übertragen.

Um eine einfache Handhabung des Systems zu gewährleisten, sollen die einzelnen Hosts zu Gruppen zusammengefasst werden können und Aktionen im Webinterface, wie das Wiederherstellen eines Abbilds, direkt auf eine gesamte Gruppe angewendet werden können.

3.7 Datenhaltung

Das System benötigt viele Informationen, um funktionstüchtig zu sein. Zum einen müssen Informationen über die Clients, sowie Informationen über die Images und die Images selbst permanent gespeichert werden. Zum anderen müssen die Informationen über die aktuell laufenden Cloningvorgänge temporär gespeichert werden. Um diese Unmenge an Daten zu speichern, eignet sich am besten eine Datenbank. Einzig zur Speicherung der Images selbst ist durch die hohen Dateigrößen eine Datenbank eher ungeeignetn und ein Dateisystem besser geeignet. Als Datenbank-Server wird MySQL eingesetzt, da es frei verfügbar ist und einen geringen Resourcenverbrauch hat.

3.8 Datenbankzugriff

Das Webservice und das Webinterface greifen direkt auf die Datenbank zu. Dabei müssen aber üblicherweise alle SQL-Abfragen händisch geschrieben werden. Um dies zu vereinfachen ist der Einsatz

[3] Asynchronous JavaScript and XML

von objektrelationaler Abbildung möglich, welche auf einer höheren Abstraktionsschicht operiert. Die objektrelationale Abbildung ermöglicht den Zugriff auf die relationale Datenbank mittels Objekten und bietet somit einen programmiersprachennahen Ansatz für den Zugriff. Diese Funktionalität wird in vielen Programmiersprachen durch Bibliotheken ermöglicht, wie beispielsweise durch Hibernate in Java.

Üblicherweise werden die Relationen der Datenbank in Objektklassen der jeweilig benutzten Programmiersprache und die Attribute der Relation durch entsprechende Attribute der Klasse dargestellt. Die einzelnen Tupel einer Relation werden als Instanzen der jeweiligen Klasse abgebildet. Das führt dazu, dass man lediglich eine Klasseninstanz im Programm anlegen und die Attribute der Instanz mit Werten belegen muss, um ein neues Tupel in eine Relation der Datenbank einzufügen. Die meisten objektrelationalen Mapper bieten eine eigene Abfragesprache an, um Abfragen aus der Datenbank zu vereinfachen. Der Entwickler muss somit nicht zwingend eine datenbankspezifische Sprache beherrschen. Wird eine Abfrage abgesetzt, kümmert sich der Mapper automatisch um die Ausführung. Das Ergebnis wird in Instanzen der jeweiligen Klasse gespeichert, welche dann sehr einfach im Programm verwendet werden können. Die Änderung von bereits bestehenden Tupel geschieht ebenfalls auf sehr unkomplizierte Weise. Es genügt, lediglich ein Attribut einer Instanz zu ändern, welche vom Mapper mit dem Tupel aus der Datenbank gefüllt wurde. Die Änderung wird dann wieder automatisch vom Mapper in die Datenbank übertragen. Meist bietet ein objektrelationaler Mapper auch diverse Möglichkeiten, um die Arbeit mit einer Datenbank zu erleichtern. Dazu zählt zum Beispiel automatische die automatische Auflösung von Relationsbeziehungen, welche meist einfach mit dem Zugriff des Entwicklers auf das Attribut einer Instanz, welches eine Beziehung mit einer anderen Relation hat, ausgeführt wird. Alle zugehörigen Tupel der in Beziehung stehenden Relation werden vom Mapper in Instanzen der entsprechenden Klasse abgebildet und können danach vom Entwickler verwendet werden.

3.9 Betriebssystemauswahl

Da das System möglichst keine Lizenzkosten verursachen soll, ist es auch erstrebenswert ein freies Betriebssystem einzusetzen. Dazu bietet sich das weit verbreitete GNU/Linux an. Für den Server wird vorerst nur die Distribution Ubuntu Linux in der Version 8.04 Hardy Heron unterstützt, da mit dem Einsatz dieser Distribution bereits die meisten Erfahrungen gemacht wurden. Die bereits etwas ältere Version 8.04 wurde deshalb gewählt, da sie Long-Term-Support mit fünf Jahren Sicherheitsupdates bietet. Im späteren Verlauf sollen auch andere Ubuntu Versionen, sowie eventuell andere Linux-Distributionen unterstützt werden.

3.10 Programmiersprachenauswahl

Für die Auswahl der Programmiersprache wurde eine Übersicht von möglichen Programmiersprachen aufgestellt. Die Programmiersprache C# wurde nicht in die Übersicht aufgenommen, da als Serverbetriebssystem nur GNU/Linux unterstützt werden soll. Obwohl mit Mono eine freie Implementierung des .NET Frameworks existiert, erscheint es nicht sinnvoll eine für das Windows Betriebssystem entwi-

ckelte Plattform unter Linux einzusetzen. Somit bleiben die Programmiersprachen Java, PHP, Python und Ruby, welche in Tabelle 3.1 aufgelistet sind.

	Java	**PHP**	**Python**	**Ruby**
Typ	Bytecode	Interpretiert	Interpretiert	Interpretiert
Erfahrung	+++++	++++	+	+
Speicherplatzbedarf der Runtime	~ 100 MiB	~ 6 MiB	~ 15 MiB	~ 6 MiB
SOAP Implementierung	verfügbar	verfügbar	verfügbar	verfügbar
Objekt-Relational Mapper	verfügbar	verfügbar	verfügbar	verfügbar

Tabelle 3.1: Vergleich der Programmiersprachen

Die Entscheidung für die Programmiersprache wurde für das Webinterface getrennt von Clientsoftware und Webservice getroffen, da es sich bei Clientsoftware und Webservice um CLI[4] Programme und beim Webinterface um eine Webseite handelt. Eine weitere Unterteilung von Clientsoftware und Webservice wurde nicht getroffen, da drei unterschiedliche Programmiersprachen zusätzlichen Aufwand bedeutet hätten und das Projekt unnötigerweise unübersichtlich gemacht hätten.

Aufgrund des hohen Speicherplatzbedarfs der Java Runtime, welche auf jedem Client zur Verfügung stehen und dadurch bei jedem Starten zum Client übertragen werden müsste, wurde Java für die Clientsoftware und das Webservice ausgeschlossen und die Auswahl beschränkte sich nun auf die Skriptsprachen PHP, Python und Ruby, welche sehr viel weniger Speicherplatz benötigen. DI (FH) Georg Ungerböck empfahl für den Einsatz mit Webservices die Programmiersprache Python und riet eher von PHP für die Webservice Implementierung ab. Nach eigenen Recherchen zu den Programmiersprachen Python und Ruby wurde klar, dass die SOAP Implementierung von Python mit dem Namen Zolera SOAP Infrastructure (ZSI) eindeutig besser geeignet ist, da die einzige SOAP Implementierung SOAP4R für Ruby anscheinend Probleme mit komplexen Datentypen hat. [Wil09] Außerdem scheint die Entwicklung von ZSI, nach dem Subversion Repository zu urteilen, sehr aktiv zu sein. Weiters wurde für den Datenbankzugriff unter Python ein objektrelationaler Mapper namens SQLAlchemy gefunden, welcher durch seinen einfachen Einsatz und seinen großen Funktionsumfang sehr vielversprechend war. Vorteilhaft ist, dass die für den Client benötigte Unterprozess-Steuerung in Python sehr einfach mit dem `subprocess`-Modul möglich ist. Ein weiterer Vorteil der Skriptsprache Python ist die einfache Syntax, welche auf wenig Formalismus optimiert ist und durch zwingendes Einrücken der Codezeilen auch automatisch zu besserer Lesbarkeit führt.

Weiters wurde entschieden das Webinterface, welches ohne SOAP Bibliothek auskommt, in der Programmiersprache PHP zu schreiben, da die meiste Erfahrung mit Webanwendungen in PHP vorhanden war. Für PHP wurde als objektrelationaler Mapper Doctrine ausgewählt.

3.11 Blockschaltbild

Das Blockschaltbild (Abbildung 3.1) zeigt den Aufbau des selbst entwickelten Cloningsystems OpenClone mit den auf das Server- und Client-System verteilten Softwarekomponenten. Auf dem Server

[4] Command Line Interface

laufen die Komponenten OpenClone Webinterface und OpenClone Webservice, wobei am Client die Clientsoftware OpenClone Engine läuft. Weiters wird in der Abbildung die Zusammenarbeit der Softwarekomponenten und deren Abhängigkeiten von anderer Software dargestellt.

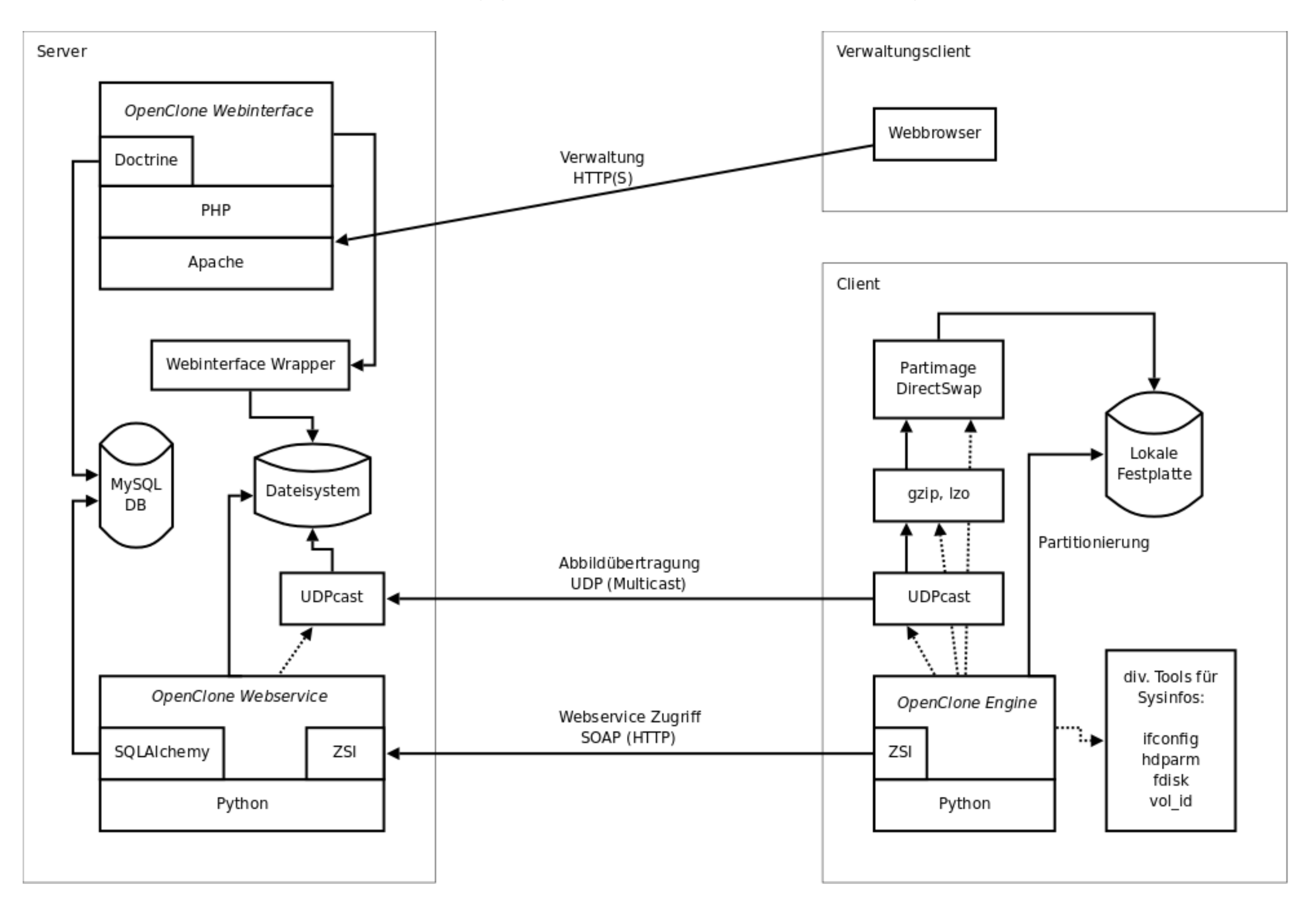


Abbildung 3.1: Blockschaltbild von OpenClone

Kapitel 4

Anwendungsentwicklung

In diesem Kapitel wird beschrieben, wie und mit welchen Hilfsmitteln die einzelnen Softwareteile entwickelt wurden und welche Herausforderungen dabei zu lösen waren. Weiters wird auf die Funktionsweise einiger Teile genauer eingegangen.

4.1 Entwicklungsumgebung

In diesem Abschnitt werden die zur Softwareentwicklung eingesetzten Entwicklungsumgebungen kurz beschrieben.

4.1.1 MySQL Datenbank

Für den EER-Entwurf der Datenkbank wurde das Tool MySQL Workbench in der Version 5.1 Beta in der Community Edition verwendet[1]. Mit diesem Werkzeug ist es möglich, eine Datenbank grafisch zu entwerfen und daraus die erforderlichen DDL-Statements für die Tabellen und deren Attribute der Datenbank automatisch zu erzeugen.

Die Software bietet unter anderem folgende nützliche Funktionen:

- Importieren von Dateien, welche DDL-Statements enthalten und aus welchen ein Diagramm erstellt wird
- Entwurf einer gesamten Datenbank in grafischer Darstellung mit Unterstützung für Schlüsselvergabe, Beziehungen und andere Funktionen, wie etwa Constraints
- Beziehungsdarstellung in der Crown's Foot Form und seit der Version 5.1 Beta auch in der klassischen EER Form
- Automatische Erzeugung einer Datei, welche die erforderlichen DDL-Statements enthält, um die entworfene Datenbank in einen MySQL Server zu importieren

[1] Frei verfügbar unter `http://www.mysql.de/products/workbench/`

Weiters wurde auf dem Server, auf dem der MySQL Serverdienst mit der Datenbank vorhanden war, das webbasierte Werkzeug phpMyAdmin[2] verwendet, um schnell und unkompliziert Änderungen an der laufenden Datenbank vornehmen zu können. Dieses Tool benötigt jedoch einen installierten Webserver auf dem System.

4.1.2 Python

Für die Entwicklung der OpenClone Engine und des Webservices in Python wurde die IDE Eclipse[3] in der Version 3.4.2 mit der Pydev[4] Erweiterung in der Version 1.4.5 für Python Unterstützung und der Subclipse[5] Erweiterung in der Version 1.4.8 für Subversion-Unterstützung verwendet.

4.1.3 PHP

Für die Entwicklung des OpenClone Webinterfaces in PHP wurde die IDE NetBeans von Sun Microsystems in der Version 6.5.1 verwendet[6]. Dabei wurde jene Version eingesetzt, welche lediglich PHP-Entwicklung unterstützt. Diese Version bietet unter anderem folgende nützliche Funktionen:

- Syntax Highlighting
- Vorschläge von Statements
- Kurzbeschreibungen zu Statements und Links zu ausführlicheren Dokumentationen auf der PHP Webseite
- Automatische Codevervollständigung
- Automatische Bereitstellung der PHP Dateien auf einem lokal installiertem Webserver
- Integrierte Subversion-Werkzeuge

Diese Features ermöglichen ein deutlich bequemeres Arbeiten mit PHP-Dateien, als mit dem zu Beginn verwendeten Texteditor gedit unter Ubuntu Linux.

4.2 Datenbank

Die Datenbank stellt die Schnittstelle zum Informationsaustausch zwischen dem OpenClone Webinterface, also der Software zum Steuern der Cloningvorgänge durch den Benutzer, und dem Webservice von OpenClone dar. Diese Datenbank wurde in MySQL von Sun Microsystems realisiert. Alle angelegten Tabellen verwenden den Typ InnoDB, um Fremdschlüsselunterstützung zu gewährleisten. In Abbildung 4.1 ist das EER-Diagramm der Datenbank dargestellt.

[2]Frei verfügbar unter `http://www.phpmyadmin.net/`
[3]Frei verfügbar unter `http://www.eclipse.org/`
[4]Frei verfügbar unter `http://pydev.sourceforge.net/`
[5]Frei verfügbar unter `http://subclipse.tigris.org/`
[6]Frei verfügbar unter `http://www.netbeans.org/`

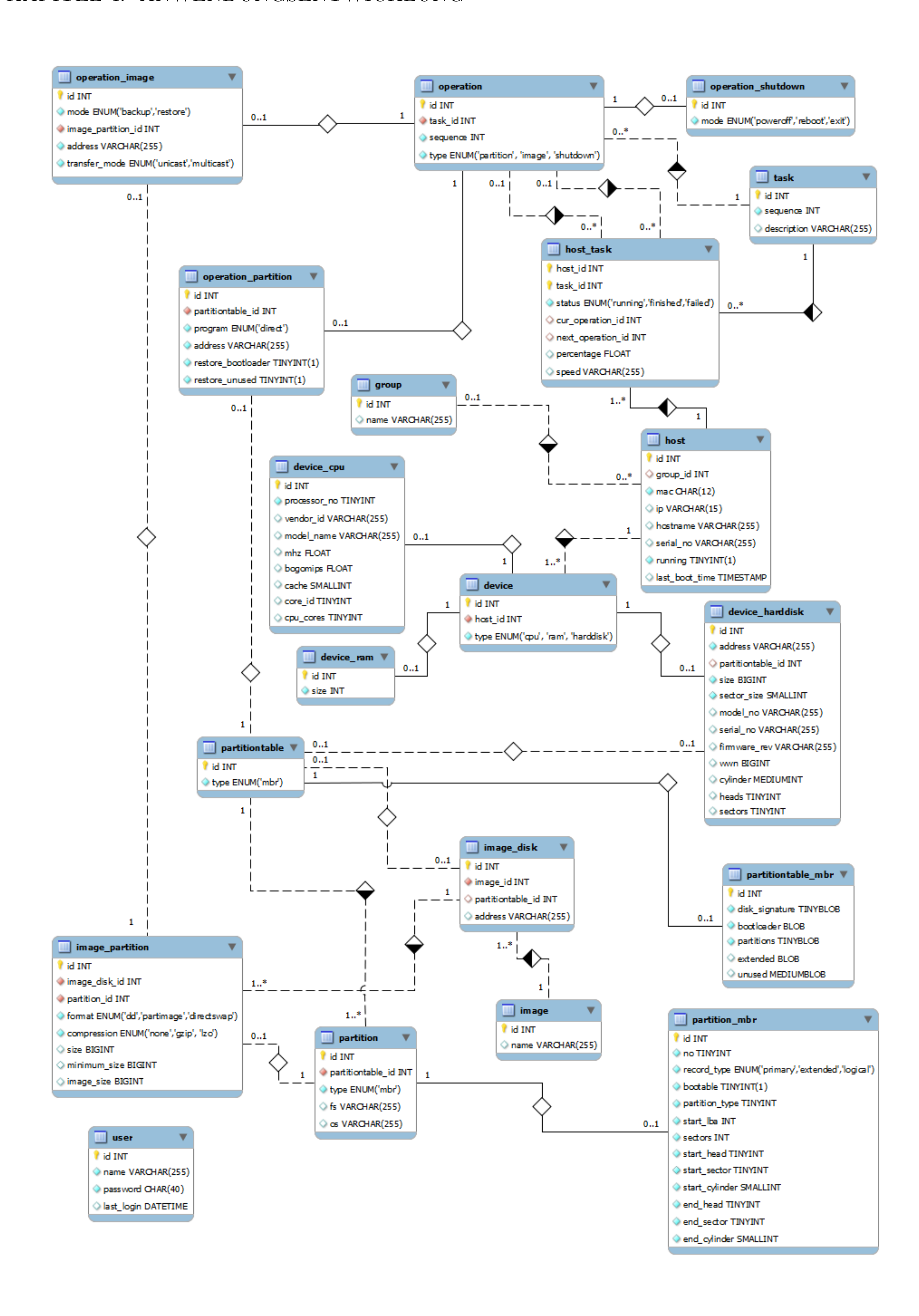


Abbildung 4.1: EER-Diagramm der Datenbank

Beim Entwurf der Datenbank wurde sehr auf eine spätere Erweiterbarkeit Wert gelegt und versucht, datenbankseitig bereits jetzt zukünftige Features, wie z.B. Images einzelner Partition oder mehrerer Festplatten, vorzusehen.

Aufbau

In der Datenbank werden mehrere Operationen (operation) zu Tasks (task) zusammengefasst und diese Tasks können auf mehreren Hosts (host_task) gleichzeitig ausgeführt werden. Die Tabelle Operation dient als Basis für alle speziellen Operationen (operation_partition, operation_image, operation_shutdown). Somit ist es später möglich, weitere Operationen zu implementieren. Die Hosts (host) können einer Gruppe (group) zugeordnet werden und jeder Host kann mehrere Devices (device) haben. Ein Device (device) kann eine CPU (device_cpu), RAM (device_ram) oder Festplatte (device_harddisk) sein. Auch hier können später auf einfache Weise weitere Devicetypen hinzugefügt werden. Jedes Image (image) kann aus mehreren Festplatten (image_disk) bestehen und jede Festplatte aus mehreren Partitionen (image_partition). Dabei hat jede Festplatte in einem Image (image_disk) und in einem Host (device_harddisk) eine Partitionstabelle (partitiontable) mit Partitionen (partition). Jede Partitionstablle und Partition kann zur Zeit nur vom Typ MBR sein, in Zukunft ist aber noch eine Implementierung der EFI Partitionstabelle vorgesehen. Eine Image-Operation (operation_image) hat nur eine Beziehung zur Partition eines Images, d.h. es muss beim Backup und Restore für jede Partition eine eigene Image-Operation angelegt werden.

4.3 Engine

Die Engine ist jener Softwareteil, der am Client gestartet wird und sich um die Ausführung der Cloningoperationen kümmert. Die Engine wurde in der Programmiersprache Python implementiert und ist nur für den Betrieb auf einem Linux System geeignet. Im Regelfall wird die Engine in einem über PXE auf dem Client gebootetem Ubuntu Linux System ausgeführt. (siehe 4.6.2)

Die Engine verbindet sich beim Starten mit dem OpenClone Webservice und überträgt daher die aktuellen Systeminformationen (logon). Wenn der Login fehlschlägt, weil der Client noch nicht registriert ist, startet die Engine die automatische Registrierung (register). Nachdem die Engine erfolgreich beim Webservice eingeloggt ist, fragt sie nach einer neuen Operationen (nextOperation) und führt diese im Anschluss aus. Sobald die Operation fertig ausgeführt wurde, wird der neue Status zum Webservice übertragen (statusUpdate) und eine neue Operation angefordert (nextOperation). Dieser Ablauf wird in der Abbildung 4.2 grafisch verdeutlicht.

Folgende Operationen werden unterstützt:

- Paritionierung (PartitionOperation)
- Partitionssicherung (ImageOperation)
- Abschalten des Rechners (ShutdownOperation)

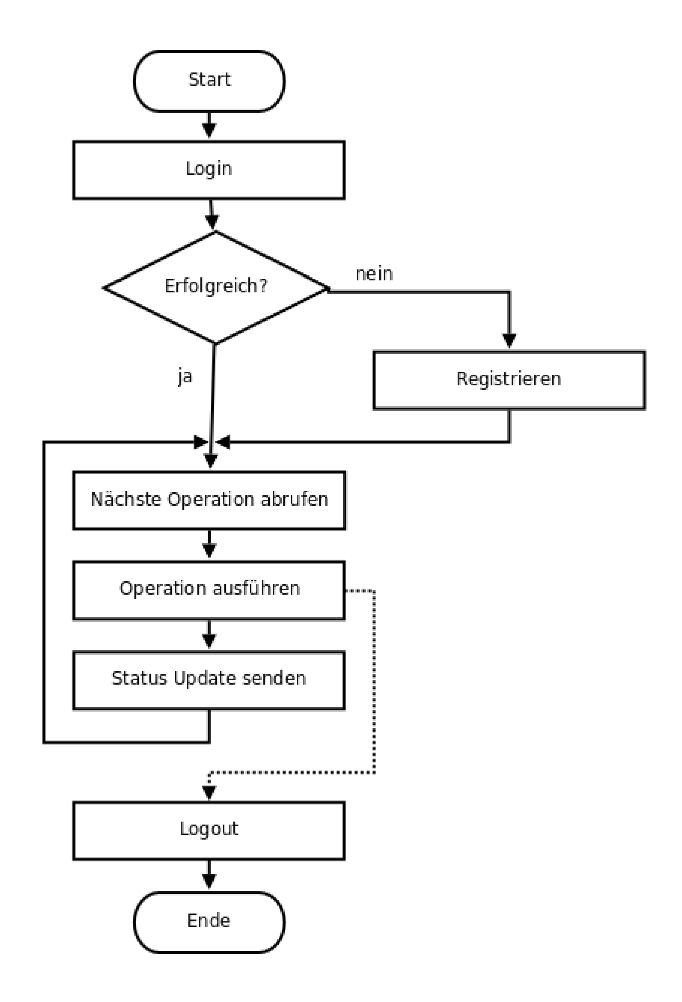


Abbildung 4.2: Ablaufdiagramm der OpenClone Engine

4.3.1 Modulübersicht

Im Folgenden werden die einzelnen Module beschrieben:

- **openclone.engine**
  - **main** benutzt das Modul *client* für die Kommunikation mit dem Webservice, beinhaltet die Hauptschleife und bearbeitet die geforderten Operationen
  - **client** stellt die Schnittstelle zum Webservice dar und setzt dazu das *engineservice* Modul ein, holt die benötigten Systeminformationen über das Modul *sysinfo* ein
  - **exception** beinhaltet die eigenen Exceptionklassen
- **openclone.common**
  - **nettools** stellt einige Netzwerktools zur Verfügung (z.B. Routing Table Lookup)
  - **utils** stellt einige Hilfsmittel bei der Kommunikation mit Prozessen zur Verfügung
  - **version** beinhaltet die Programmversion
- **openclone.engine.sysinfo**

  - **sysinfo** sammelt die benötigten Systeminformationen von verschiedenen Quellen (siehe 4.3.3)
  - **ioctl** beinhaltet Hilfsfunktionen für ioctl
  - **blockdev** beinhaltet die ioctl Kommandos für Block Geräte

- **openclone.engine.partition** beinhaltet alle implementierten Partitionierungsprogramme (siehe 4.3.4)
  - **directpart** kümmert sich um das Auslesen und Schreiben der Festplattenpartitionierung

- **openclone.engine.imaging** beinhaltet alle implementierten Sicherungsprogramme (siehe 4.3.5)
  - **partimage** stellt eine Schnittstelle zum Sicherungsprogramm Partimage zur Verfügung
  - **directswap** implementiert die Sicherung von Linux Swap Partitionen

- **openclone.engine.compression** beinhaltet alle implementierten Kompressionsverfahren (siehe 4.3.5)
  - **gzip** stellt eine Schnittstelle zum Kompressionsprogramm gzip zur Verfügung
  - **lzo** stellt eine Schnittstelle zum Kompressionsprogramm lzop zur Verfügung

- **openclone.common.transfer** beinhaltet alle implementierten Übertragungsmethoden (siehe 4.3.5)
  - **udpcast** stellt eine Schnittstelle und Verwaltungstools für das Übertragungsprogramm UDPcast zur Verfügung

- **openclone.engine.misc**
  - **hash** stellt Hashfunktionen (z.B. sha1) zum Debuggen der Übertragung zur Verfügung

- **openclone.engine.webservice**
  - **engineclient** vereint alle benötigten Klassen, Methoden für das Webservice
  - **OpenCloneEngineService_services_types** stellt die vom Webservice eingesetzten komplexen Datentypen als Klassen zur Verfügung
  - **OpenCloneEngineService_services** stellt die Methoden zum Aufruf des Werbservices zur Verfügung

4.3.2 Abhängigkeiten

Um alle Funktionalitäten der Engine nutzen zu können, müssen folgende Abhängigkeiten zur Verfügung stehen:

- Python 2.5 (python)
- Zolera SOAP Infrastructure 2.0 (python-zsi)

- UDPcast
- Partimage >= 0.6.6
- gzip
- lzop
- dmidecode
- hdparm >= 9.2 (siehe 4.3.3)
- vol_id (udev)
- ifconfig (net-tools)
- fdisk, mkswap (util-linux)
- shutdown

4.3.3 Systeminformationen

Beim Registrieren und Einloggen werden die aktuellen Systeminformationen zum OpenClone Webservice übertragen. Diese Systeminformationen sind teilweise zur Funktion des Cloningsystems notwendig (z.B. Netzwerkschnittstelle, Festplatten, Partitionen) und teilweise nur zur Information des Benutzers im Webinterface (z.B. Prozessor, Arbeitsspeicher) notwendig. Aus welchen Quellen diese Informationen gewonnen werden, zeigt die Tabelle 4.1.

Information	**Quelle**
Prozessoren	Datei /proc/cpuinfo
Arbeitsspeicher	Datei /proc/meminfo
Netzwerkschnittstelle	Externes Programm /sbin/ifconfig
Partitionen	Datei /proc/partitions
Festplattendetails	Direkt über ioctl, Externes Programm /sbin/hdparm
Partitionsdetails	Externes Programm /sbin/fdisk
Dateisystem	Externes Programm /lib/udev/vol_id
Master Boot Record, Extended Boot Record	Direkter Zugriff (siehe 4.3.4)
Seriennummer	Externes Programm /usr/sbin/dmidecode

Tabelle 4.1: Verwendete Systeminformationen und deren Quellen

Zum Auslesen des WWN (World Wide Name) der Festplatten, der weltweit eindeutigen Kennung, wird das Programm `hdparm` verwendet. Dieses gibt den WWN jedoch vor der Version 9.2 in einem falschen Format aus, sodass er nicht mehr eindeutig ist. In der am Client eingesetzten Ubuntu Liunx Version 8.10 Hardy Heron wird die `hdparm` Version 8.9 mitgeliefert, somit ist mit dem aktuellen System das

Auslesen des WWN nicht immer eindeutig möglich. Eine alternative Methode wäre den WWN direkt über den `HDIO_GET_IDENTITY` ioctl Befehl auszulesen. Diese Methode wurde jedoch nicht umgesetzt, da sie viel komplexer ist und der WWN nicht für die Funktion des Cloningsystems benötigt wird, sondern nur zur Information des Benutzers dient.

4.3.4 Partitionierung

Die Partitionierung ist die Aufteilung der Festplatte in mehrere logische Teile. Sie wird beim Abrufen der Systeminformationen ausgelesen und kann mittels einer PartitionOperation geschrieben werden. Das Lesen und Schreiben der Partitionierung wurde mit dem Modul *directpart* zur Gänze selbst in Python implementiert. Es werden das weit verbreitete MBR[7]- und EBR[8]-Format und damit sowohl primäre, als auch erweiterte Partitionen mit logischen Partitionen unterstützt. Das neueste Partitionsformat EFI[9] wurde allerdings aufgrund seiner Komplexität und seiner geringen Verbreitung noch nicht implementiert.

Da die Partitionierung beim Einloggen als Teil der Systeminformationen übertragen wird, ist beim Erstellen der Sicherung einer Festplatte kein explizites Auslesen mittels einer Operation notwendig. Nur beim Wiederherstellen einer Sicherung wird eine PartitionOperation zum Schreiben der Partitionierung benötigt.

Die Tabelle 4.2 und die Tabelle 4.3 zeigen, wie ein MBR und EBR aufgebaut ist und welche Teile davon gesichert werden.

Anfang	Bezeichnung	Größe
0	Bootloader	440 Byte
440	Disk Signatur	4 Byte
444	Unbenutzt (0x00)	2 Byte
446	1. primärer Partitionseintrag	16 Byte
462	2. primärer Partitionseintrag	16 Byte
478	3. primärer Partitionseintrag	16 Byte
494	4. primärer Partitionseintrag	16 Byte
510	MBR Signatur (0xAA55)	2 Byte

Tabelle 4.2: MBR Aufbau (Gesamtgröße 512 Byte)[10][GS09]

Dabei werden absichtlich auch die unbenutzte Teile des MBR und EBR gesichert, um die Kompatibilität zu erhöhen. Die Linux Distribution SuSE installierte beispielsweise auf einem Testrechner mit Windows Vista den Bootloader GRUB automatisch in die erweiterte Partition und markierte diese als aktiv. Mehrere Quellen beschreiben die ersten 446 Byte des EBR als normalerweise unbenutzt. Es wird dabei aber nur auf einen alten IBM Bootloader hingewiesen, welcher einige Bytes für den Menüeinträge der

[7]Master Boot Record
[8]Extended Boot Record
[9]Extensible Firmware Interface
[10]Die grau hinterlegten Bereiche werden nicht in die Sicherung miteinbezogen.
[11]Die grau hinterlegten Bereiche werden nicht in die Sicherung miteinbezogen.

Anfang	Bezeichnung	Größe
0	Unbenutzt (0x00)	446 Byte
446	Logischer Partitionseintrag	16 Byte
462	Nächster EBR	16 Byte
478	Unbenutzt (0x00)	32 Byte
510	MBR Signatur (0xAA55)	2 Byte

Tabelle 4.3: EBR Aufbau (Gesamtgröße 512 Byte)[11][GS09]

Partition benutzt, jedoch nicht dass der Bereich auch manchmal für Bootloader Code genutzt wird. [AA09, GS09, Sed09]

Sichern von Bootloadern

Eine spezielle Herausforderung gab es beim Sichern von manchen Bootloadern, die im MBR installiert werden. Um dieses Problem besser erklären zu können, ist in der Tabelle 4.4 anhand eines Beispiels die üblichen Belegung einer Festplatte dargestellt. Im Beispiel sind auf der Festplatte zwei primäre Partitionen mit je ungefähr 3,1 MiB angelegt.

Anfang	Bezeichnung	Größe
0	MBR	1
1	Unbenutzt	62
63	1. Partition	6300
6363	2. Partition	6300

Tabelle 4.4: Beispiel für eine Festplattenbelegung (in 512 Byte Blöcken)

Diese Aufteilung ergibt sich dadurch, dass ältere Betriebssysteme nur Partitionen unterstützten, die an Zylindergrenzen beginnen und enden. [Bra09]

Normalerweise würde man annehmen, dass die Sicherung des MBR, sowie aller Partitionen reicht, damit der Bootloader bei später Wiederherstellung wieder voll funktionsfähig ist.

Leider haben Bootloader, welche in den MBR installiert werden, dort oft zu wenig Speicherplatz (440 Byte), um den gesamten Code unterzubringen. Dieses Problem umgehen sie dadurch, dass sie den eigentlich unbenutzten Speicherplatz (31 KiB) zwischen MBR und erster Partition nutzen. Der Bootloader GRUB ist ein Vertreter dieser Gattung, denn bei GRUB wird der sogenannte Stage 1.5 im unbenutzten Bereich installiert. [Bra09] Wenn nun eine Festplatte mit so einem Bootloader gesichert wird, ist der Bootloader nach dem Wiederherstellen nicht mehr funktionsfähig.

Es gibt nun zwei Möglichkeiten dieses Problem zu lösen:

1. Den Bootloader nach der Wiederherstellung von MBR und Partitionen neu installieren

2. Den unbenutzten Bereich ebenfalls sichern und wiederherstellen

Die erste Möglichkeit wird vom Cloningsystem Clonezilla eingesetzt. Dadurch ist aber nur die Wiederherstellung von bestimmten Bootloadern möglich. Bereits beim Testen mit Clonezilla wurde ein Problem mit der Neuinstallation von GRUB erkennbar, da dieser Bootloader nach der Wiederherstellung nicht mehr grafisch, sondern nur mehr textuell angezeigt wird.

Die zweite Möglichkeit wird vom Cloningsystem FOG eingesetzt, welches immer den kompletten ersten Zylinder, also den MBR inklusive dem unbenutzten Bereich sichert. Dadurch ist die Art des verwendeten Bootloaders unwichtig und führt zu keinen weiteren Problemen.

Beim OpenClone wurde ebenfalls die zweite Möglichkeit implementiert, jedoch noch etwas flexibler als bei FOG. Es wird nicht immer der gesamte erste Zylinder gesichert, sondern genau der Bereich zwischen MBR und erster Partition.

4.3.5 Partitionssicherung

Bei der Partitionssicherung wird eine gesamte Partition des Clients in eine Datei am zentralen Server gesichert. Dazu gehört in jedem Fall die Abbildung (Imaging), welche die Daten der Partition liest und in ein geeignetes Format bringt sowie umgekehrt, und die Übertragung (Transfer), welche die Daten der Abbildung vom bzw. zum zentralen Server überträgt.

Die OpenClone Engine wurde so konzipiert, dass verschiedene Implementierungen, die unterschiedliche Features haben können, zur Abbildung und zur Übertragung verwendet werden können. Beim Erstellen einer Sicherung dient die Abbildung als Quelle und die Übertragung als Ziel. Beim Wiederherstellen einer Sicherung analog die Übertragung als Quelle und die Abbildung als Ziel. Zwischen Quelle und Ziel können noch beliebig weitere Zwischenstellen eingefügt werden, wie z.B. eine Komprimierung oder eventuell eine Verschlüsselung. Zur Zeit ist jedoch nur die Komprimierung und das Hashmodul als Zwischenstelle implementiert. Die bei der Erstellung verwendete Abbildungs-, Übertragungs- und Kompressionsimplementierung wird am zentralen Server gespeichert, um später eine Wiederherstellung zu ermöglichen.

Abbildung (Imaging)

Es gibt grundsätzlich zwei verschiedene Möglichkeiten ein Abbild einer Partition zu erstellen:

- Byte für Byte und ohne Kenntnis des Dateisystems
- Intelligent und mit Hilfe des Dateisystems

Um Speicherplatz zu sparen, wird üblicherweise nicht die gesamte Partition Byte für Byte gesichert, sondern die intelligente Variante gewählt. Dabei wird über das Dateisystem der Partition die benutzten Teile festgestellt und genau diese gesichert. Dadurch ist es jedoch notwendig, den Aufbau des Dateisystems zu kennen. Deshalb nur die bekannten Dateisysteme unterstützt werden.

Die OpenClone Engine bietet aktuell zwei Implementierungen:

- **PartImage**, welche eine Schnittstelle zur Imagingsoftware Partimage darstellt

- **DirectSwap**, welche die Tools `vol_id` und `mkswap` verwendet, um Linux Swap Partitionen möglichst effizient zu sichern und dazu ein eigenes Image Format implementiert, welches im Listing 4.1 dargestellt wird

Beide Implementierungen gehen nach der intelligenten Methode vor und unterstützen unterschiedliche Dateisysteme, welche in der Tabelle 4.5 genauer aufgelistet sind. Weitere interessante Implementierungen wären Schnittstellen zu den Imagingutilities Partclone und ntfsclone, welche auch in die Liste der unterstützten Dateisysteme aufgenommen wurden.

Die Auswahl der Implementierung, welche zur Sicherung einer Partition eingesetzt wird, kann vom Webinterface aus getroffen werden. Dafür wird das Dateisystem der Partition, das in den Systeminformationen, die beim Registrieren und Einloggen gewonnen wurden, enthalten ist, verwendet.

Implementierung	**Imagingutility**	**Unterstützte Dateisysteme**
DirectSwap	-	Linux Swap
PartImage	Partimage 0.6.7	*stable:* Ext2/Ext3, ReiserFS, FAT16/32, HPFS, JFS, XFS *beta:* UFS, HFS *experimental:* NTFS
-	Partclone 0.1.0	Ext2/Ext3/Ext4, ReiserFS, Reiser4, XFS, FAT, HFS Plus, NTFS
-	ntfsclone	NTFS

Tabelle 4.5: Unterstütze Dateisysteme der verschiedenen Implementierungen[12]

DirectSwap ist eine eigene Implementierung für Linux Swap Partitionen. Sie verwendet zur Sicherung das Tool `vol_id`, um die UUID und das Label der Swap Partition auszulesen und speichert es wie im Listing 4.1 angegebenem Format. Zur Wiederherstellung der Swap Partition wird das Tool `mkswap` mit der UUID und dem Label als Parameter aufgerufen.

```
1 #OpenClone directswap file
2 UUID=<UUID>
3 LABEL=<Label>
```

Listing 4.1: DirectSwap Abbild Format

Partclone ist bei den vorgegangenen Tests mit der Version 0.0.9 noch etwas instabil gewesen, daher wurde vorerst nur eine Schnittstelle für Partimage implementiert.

ntfsclone soll in Zukunft eine weitere Alternative für NTFS Dateisysteme darstellen und den Cloningvorgang schneller machen.

[12] vgl. Partimage [Par09b]. Partclone [Par09a]. ntfsclone [LNP09]. Noch nicht erfolgte Implementierungen/Imagingutilities sind grau hinterlegt.

Komprimierung (Compression)

Die Komprimierung ermöglicht eine weitere Verringerung des benötigten Speicherplatzes für die Abbilder. Wichtig ist dabei nicht unbedingt der Speicherplatzverbrauch am zentralen Server, sondern die geringere Bandbreite, die zur Übertragung notwendig ist. So ist eine schnellere Partitionssicherung und -wiederherstellung möglich, da sich der Engpass von der Netzwerkverbindung zum Prozessor oder zur Festplatte verlagert.

Für die Komprimierung bietet die OpenClone Engine aktuell folgende Implementierungen:

- Gzip, welches eine Schnittstelle zur gzip Kompressionssoftware darstellt
- Lzo, welches eine Schnittstelle zur lzop Kompressionssoftware darstellt

Vergleich der Kompressionssoftware

Die folgenden Kompressionsutilities wurden getestet:

- gzip[13], welches den Lempel-Ziv (LZ77) Kompressionsalgorithmus verwendet. [Fre09]
- bzip2[14], welches den Burrows-Wheeler Kompressionsalgorithmus und die Huffman Kodierung verwendet. [Sew09]
- lzma[15], welches den Lempel-Ziv-Markov Kompressionsalgorithmus, eine Weiterentwicklung des Lempel-Ziv (LZ77) Algorithmus, verwendet. [Wik09a]
- lzop[16], welches den Lempel-Ziv-Oberhumer Kompressionsalgorithmus verwendet. [Obe09]

Als Testsystem wurde ein Notebook mit AMD Turion 64 X2 TL-60 2 GHz und 2 GiB Arbeitsspeicher im Netzbetrieb eingesetzt. Als Betriebsysstem diente Ubuntu Linux 8.10 Intrepid Ibex mit dem Linux Kernel 2.6.27-11-generic. Als Eingabedatei wurde ein mit der Software Partimage erstelltes Abbild einer frisch installierten Windows XP Partition verwendet. Die Eingabedatei hat eine Größe von 2993,9 MiB (3139314716 Byte).

Die Tabelle 4.6 listet die mit verschiedenen Parametern getesteten Komprimierungsutilites nach dem Kompressionsgrad auf. Am besten hat dabei lzma abgeschnitten, welches einen Kompressionsgrad von 69,22% erreichte. Der minimale Speicherplatzverbrauch wird jedoch durch eine sehr lange Kompressionszeit erkauft. Wie in der Tabelle 4.7 erkannt werden kann, schneidet lzma in der auf die Netzwerkübertragung ausgelegte Wertung durch die lange Kompressionszeit eher schlecht ab. Auf den vorderen Plätzen liegen lzop und gzip, welche deutlich schneller komprimieren als lzma. Bei der Sicherung ist lzop durch seine kurze Kompressionsdauer unschlagbar. Denn lzop ist die einzige Kompressionssoftware, die sogar weniger Zeit zur Kompression benötigt, als die Übertragung dauert. Das bedeutet, dass lzop auch für höhere Übertragungsgeschwindigkeiten als 100 MBit geeignet ist und der Engpass trotz Kompression nicht am CPU, sondern weiterhin am Netzwerk liegt.

[13] Frei verfügbar unter `http://www.gnu.org/software/gzip/`
[14] Frei verfügbar unter `http://www.bzip.org/`
[15] Frei verfügbar unter `http://www.7-zip.org/sdk.html`
[16] Frei verfügbar unter `http://www.lzop.org/`

Format[a]	Kompression CPU Zeit	Dekompression CPU Zeit	Größe (MiB)	Kompressionsgrad
lzma L9	1:12:14.7	2:26.0	921,6	69,22%
lzma L7 (Std.)	49:14.8	2:30.2	944,1	68,46%
lzma L1	9:20.9	2:42.1	1051,2	64,89%
bzip2 L9 (Std.)	16:34.9	5:09.5	1112,2	62,85%
bzip2 L6	15:21.6	4:48.7	1114,1	62,79%
gzip L9	5:32.7	0:37.3	1121,5	62,54%
gzip L6 (Std.)	3:06.2	0:37.4	1123,7	62,47%
bzip2 L1	12:36.2	3:32.6	1130,8	62,23%
gzip L1	2:11.7	0:36.3	1164,1	61,12%
lzop L9	24:05.6	0:14.4	1200,7	59,90%
lzop L7	7:15.2	0:14.2	1203,1	59,82%
lzop L1	0:54.9	0:12.7	1290,1	56,91%
lzop L3 (Std.)	0:51.6	0:12.2	1292,8	56,82%
unkomprimiert	-	-	2993,9	0,00%

[a]Die L-Werte geben den verwendeten Level der Kompression an. L1 entspricht dem niedrigsten Level und L9 dem höchsten Level. Mit Std. wird der Standardlevel gekennzeichnet.

Tabelle 4.6: Komprimierungssoftware sortiert nach Kompressionsgrad

Bei der Wiederherstellung ist gzip am schnellsten, da sowohl gzip als auch lzop schneller dekomprimieren, als die Übertragung erfolgen kann aber der ungefähr 5 Prozentpunkte höhere Kompressionsgrad von gzip zum Tragen kommt. Die Tabelle 4.8 stellt schließlich den Kompressionsgrad der Gesamtdauer gegenüber. Damit können schnell die in Frage kommenden Algorithmen herausgefunden werden. Denn man wird nur jene Kompressionsalgorithmen einsetzen, die im Gegenzug zu einer schlechteren Gesamtzeit zumindest einen besseren Kompressionsgrad bieten.

Dadurch ergibt sich folgende Reihung:

- lzop Level 1 bietet die kürzeste Gesamtdauer
- gzip Level 1 scheint die sinnvollste Wahl zu sein, da ein um 5% Prozentpunkte besserer Komprimierungsgrad durch eine minimal längere Sicherungsdauer von unter 20 Sekunden erreicht wird.
- gzip Level 6
- gzip Level 9 bietet die schnellste Wiederherstellungszeit
- lzma Level 1
- lzma Level 7
- lzma Level 9

Format	Kompr. CPU Zeit	Dekompr. CPU Zeit	Übertr. Zeit[a]	Sicher. Dauer[b]	Wiederh. Dauer[c]	Gesamt Dauer[d]
lzop L1	0:54.9	0:12.7	1:53.9	1:53.9	1:53.9	3:47.8
lzop L3 (Std.)	0:51.6	0:12.2	1:54.2	1:54.2	1:54.2	3:48.3
gzip L1	2:11.7	0:36.3	1:42.8	2:11.7	1:42.8	3:54.5
gzip L6 (Std.)	3:06.2	0:37.4	1:39.2	3:06.2	1:39.2	4:45.4
gzip L9	5:32.7	0:37.3	1:39.0	5:32.7	1:39.0	7:11.8
unkomprimiert	-	-	4:24.4	4:24.4	4:24.4	8:48.7
lzop L7	7:15.2	0:14.2	1:46.2	7:15.2	1:46.2	9:01.5
lzma L1	9:20.9	2:42.1	1:32.8	9:20.9	2:42.1	12:03.0
bzip2 L1	12:36.2	3:32.6	1:39.9	12:36.2	3:32.6	16:08.8
bzip2 L6	15:21.6	4:48.7	1:38.4	15:21.6	4:48.7	20:10.3
bzip2 L9 (Std.)	16:34.9	5:09.5	1:38.2	16:34.9	5:09.5	21:44.4
lzop L9	24:05.6	0:14.4	1:46.0	24:05.6	1:46.0	25:51.6
lzma L7 (Std.)	49:14.8	2:30.2	1:23.4	49:14.8	2:30.2	51:45.0
lzma L9	1:12:14.7	2:26.0	1:21.4	1:12:14.7	2:26.0	1:14:40.7

[a]**Übertragungszeit:** Für die Berechnung wurde eine Übertragungsgeschwindigkeit von 95 MBit angenommen. Dieser Wert beruht auf dem gemessenen Wert einer UDPcast Übertragung über ein 100Base-TX Netzwerk von 94,5 MBit und der durch Overhead limitierten Obergrenze bei UDPcast Übertragungen von 95,8 MBit.

[b]**Sicherungsdauer:** Entspricht der kürzeren Zeit aus Kompression und Übertragung, da die Daten gleichzeitig komprimiert und übertragen werden.

[c]**Wiederherstellungsdauer:** Entspricht der kürzeren Zeit aus Dekompression und Übertragung.

[d]**Gesamt Dauer:** Entspricht der Summe der Sicherungs- und Wiederherstellungsdauer.

Tabelle 4.7: Komprimierungssoftware sortiert nach Dauer einer Sicherung und Wiederherstellung

Die Kompressionssoftware lzop im Level 1 ist vor allem für schnellere Netzwerkverbindungen geeignet, daher wurde sie inzwischen ebenfalls implementiert. gzip im Level 1 ist gut für ein 100 MBit Netzwerk geeignet und wurde daher auch als erstes Verfahren implementiert. Mit höheren Leveln kann bei gzip durch höhere Sicherungszeiten eine minimal besserer Kompressionsgrad erkauft werden, wobei die Wiederherstellungszeiten sogar minimal niedriger werden. Bei lzma kann durch Akzeptanz einer Erhöhung der Sicherungs- und Wiederherstellungszeit, je nach Level eine weitere Verbesserung des Kompressionsgrad um nahezu 10 Prozentpunkte herausgeholt werden. Die Kompressionssoftware bzip2 ist für die Komprimierung von Abbildern bei gleichzeitiger Übertragung eher ungeeignet und fällt daher sogar komplett aus der Liste.

Übertragung (Transfer)

Zur Übertragung wird die Software UDPcast[17] eingesetzt. Diese bietet die Möglichkeit mittels Multicast Übertragungen gleichzeitig an mehrere, aber nicht zwingend an alle Rechner im Netzwerk zu senden. Dadurch wird die Performance verbessert, da keine separate Übertragung der Daten zu jedem Rechner, wie bei Unicast notwendig ist und gleichzeitig die Übertragung nicht an alle Rechner im Netzwerk gehen

[17]Frei verfügbar unter `http://udpcast.linux.lu/`

Format	Kompr. CPU Zeit	Dekompr. CPU Zeit	Größe (MiB)	Kompres-sionsgrad	Gesamt Dauer
lzop L1	0:54.9	0:12.7	1290,1	56,91%	3:47.8
lzop L3 (Std.)	0:51.6	0:12.2	1292,8	56,82%	3:48.3
gzip L1	2:11.7	0:36.3	1164,1	61,12%	3:54.5
gzip L6 (Std.)	3:06.2	0:37.4	1123,7	62,47%	4:45.4
gzip L9	5:32.7	0:37.3	1121,5	62,54%	7:11.8
unkomprimiert	-	-	2993,9	0,00%	8:48.7
lzop L7	7:15.2	0:14.2	1203,1	59,82%	9:01.5
lzma L1	9:20.9	2:42.1	1051,2	64,89%	12:03.0
bzip2 L1	12:36.2	3:32.6	1130,8	62,23%	16:08.8
bzip2 L6	15:21.6	4:48.7	1114,1	62,79%	20:10.3
bzip2 L9 (Std.)	16:34.9	5:09.5	1112,2	62,85%	21:44.4
lzop L9	24:05.6	0:14.4	1200,7	59,90%	25:51.6
lzma L7 (Std.)	49:14.8	2:30.2	944,1	68,46%	51:45.0
lzma L9	1:12:14.7	2:26.0	921,6	69,22%	1:14:40.7

Tabelle 4.8: Überblick Komprimierungssoftware nach Dauer einer Sicherung und Wiederherstellung

muss, wie bei Broadcast. Somit wird nur die minimal benötigte Bandbreite des Netzwerks belegt. Da TCP[18] nur für Punkt-zu-Punkt Verbindungen geeignet ist, setzt UDPcast das Transportprotokoll UDP[19] ein. Um die Übertragung sicherzustellen, implementiert UDPcast eine eigene Sicherungsschicht, wobei die Bestätigungen der Empfänger an den Sender über Unicast gesendet werden. Außerdem implementiert UDPcast so genannte Slices, damit nicht jedes Paket einzeln bestätigt werden muss, ähnlich dem Windowing bei TCP. UDPcast wartet mit der Übertragung neuer Daten so lange, bis alle Empfänger die Daten erhalten haben bzw. wiederholt die alten Daten, falls sie verloren gehen. Dadurch orientiert sich UDPcast bei der Übertragungsgeschwindigkeit am langsamsten Empfänger. Wiederholte Übertragungen gehen ebenfalls über Multicast an alle Empfänger. [UDP09]

Ein Nachteil von Multicast ist, dass es von allen Netzwerkgeräten, also von den Rechnern, Switches und Routern, die an der Übertragung beteiligt sind, unterstützt werden muss. Um den Multicast Übertragungen zu verwalten wird IGMP[20] eingesetzt. Die Rechner können mit IGMP Join Nachrichten Multicast Gruppen beitreten und mit IGMP Leave Nachrichten Multicast Gruppen verlassen. Diese IGMP Nachrichten werden von den Multicast Routern verarbeitet und die Multicast Übertragungen an die entsprechenden Netzwerksegmente weitergeleitet. Die Switches in diesem Netzwerksegment leiten die Multicast Übertragungen normalerweise wie Broadcasts an alle Rechner weiter. Um auch im Netzwerksegment die notwendige Bandbreite zu optimieren, wird von vielen Switches IGMP Snooping unterstützt. Dabei werden die IGMP Nachrichten der Rechner auch vom Switch verarbeitet. Dieser kann dadurch die Mutlicast Übertragungen gezielt an jene Rechner weiterleiten, die der Multicast Gruppe angehören.

[18]Transmission Control Protocol
[19]User Datagram Protocol
[20]Internet Group Management Protocol

Die notwendigen Schritte zur IGMP Konfiguration auf diversen HP Switches werden im B.3.3 beschrieben.

Eine Übertragung mit UDPcast kann immer nur innerhalb eines Subnetzes erfolgen. Daher muss sobald mehrere Subnetze auf einer Netzwerkschnittstelle oder mehrere Netzwerkschnittstellen verwendet werden, die Entscheidung getroffen werden, welche Schnittstelle mit welcher IP-Adresse verwendet wird. Dazu wird die Routing Tabelle des Linux Betriebssystems (/proc/net/route) und das Programm `ip` aus dem iproute2 Paket zur Hilfe genommen.

Vorgehensweise bei der Schnittstellen- und IP Auswahl:

1. Ermitteln der Client IP Adresse über den Webserver des Webservices
2. Finden von Routen, deren Ziel die Client IP enthält
3. Finden jener Routen, welche die größte Netmask besitzen
4. Auswahl der Route, welche die kleinste Metrik besitzt
5. Ermitteln der IP-Adresse der Schnittstelle im Zielnetz

4.3.6 Abschalten des Rechners

Diese Funktion wird mit der ShutdownOperation implementiert. Dabei gibt es drei unterschiedliche Varianten der Operation:

- **shutdown** - Der Rechner wird heruntergefahren und wenn möglich ausgeschaltet
- **reboot** - Der Rechner wird neu gestartet
- **exit** - Die Software wird beendet

Für die shutdown und reboot Variante wird das Programm `shutdown` mit unterschiedlichen Parmetern verwendet.

4.4 Webservice

Das Webservice stellt die Schnittstelle zwischen der Engine und der Datenbank sowie dem Dateisystem des Servers dar. Die Kommunikation erfolgt mit dem SOAP Protokoll, welches auf der Webservice Architektur beruht. Das Webservice wurde in Python implementiert und verwendet die Zolera SOAP Infrastructure (ZSI)[21], eine Implementierung des SOAP Protokolls, und den objektrelationalen Mapper SQLAlchemy[22] zum Zugriff auf die Datenbank. Der Einsatz der Zolera SOAP Infrastucture wird im Kapitel 4.4.4 noch näher erklärt und der Einsatz von SQLAlchemy im Kapitel 4.4.5.

[21] Frei verfügbar unter `http://pywebsvcs.sourceforge.net/`
[22] Frei verfügbar unter `http://www.sqlalchemy.org/`

4.4.1 Webserviceentwurf

Das Webservice wird durch eine WSDL[23] Datei beschrieben. Diese beinhaltet alle Methoden des Webservices und kann auch die dazugehörigen Datentypen beinhalten.

Das Webservice wurde hauptsächlich mit Hilfe des grafischen Designers der Entwicklungsumgebung Eclipse entworfen, welcher die Arbeit erleichtert, jedoch trotzdem Kenntnisse über WSDL fordert.

Aufbau einer WSDL Datei[24]

- **Service:** Jedes Webservice hat einen Namen und kann eine oder mehrere Endpunkte mit einer Adresse beinhalten, wobei jeder Endpunkt ein bestimmtes Binding hat.
- **Binding:** Jedes Binding hat einen Schnittstellentyp mit Anforderungen an ein bestimmtes Datenformat.
- **Schnittstellentyp:** Jeder Schnittstellentyp beinhaltet bestimmte Operationen mit den benötigten Ein- und Ausgabenachrichten.
- **Nachricht:** Jede Nachricht kann mehrere benannte Teile enthalten, die ein bestimmtes Element enthalten.
- **Element:** Elemente haben einen Namen sind von einem bestimmten einfachen oder komplexen Datentyp.
- **Attribut:** Attribute haben einen Namen und sind von einem bestimmten einfachen Datentyp.
- **Datentyp:** Es gibt einfache und komplexe Datentypen. Einfache Datentypen sind z.B. Strings und Integer-Zahlen. Komplexe Datentypen können aus Attributen und wiederum Elementen aufgebaut sein. Weiters können Datentypen von anderen Datentypen abgeleitet werden.

Eingehaltene Regeln für einen ZSI kompatiblen Webserviceaufbau:

- Jedes Binding ist vom Style `document`
- Jede Operation besitzt eine Ein- und Ausgabenachricht und keine Fehlernachricht
- Jede Nachricht setzt sich aus einem einzigen Teil zusammen, daher entspricht sie einem Parameter der Operation bei Eingabenachrichten oder einem Rückgabewert bei Ausgabenachrichten
- Ein einer Nachricht zugeordnetes Element beinhaltet selbst nur andere Elemente und keine Attribute
- Den Datentyp Base64String vermeiden

[23] Web Service Description Language
[24] [Wik09b]

Diese Regeln waren leider nicht im Vorhinein klar und mussten erst mühsam durch viele Tests und Debuggen von ZSI erarbeitet werden, da die Dokumentation von ZSI in manchen Bereichen nicht sehr umfangreich ist. Weiteres dazu im Kapitel 4.4.4.

Obwohl ZSI nur einteilige Nachrichten unterstützt, können trotzdem über die Unterstützung von komplexen Datentypen mehrparametrige Methoden, wie auch im OpenClone Webservices verwendet, erreicht werden.

Die Abbildung 4.3 bietet eine Übersicht über das Webservice mit Service, Binding, Schnittstellentyp, Operationen, Nachrichten und den zugehörigen Elementen.

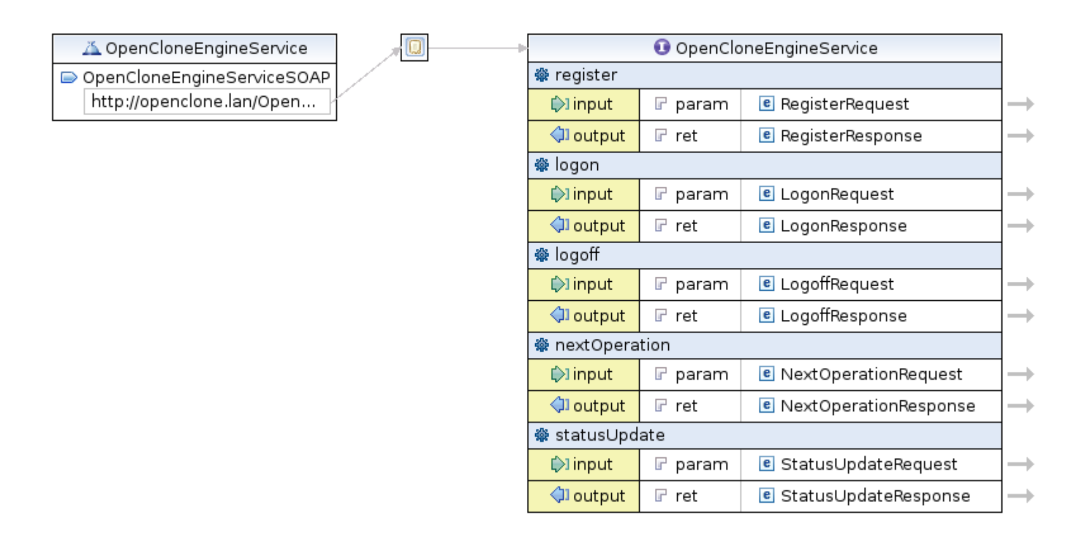


Abbildung 4.3: Webservice Übersicht (WSDL)

In der Abbildung 4.4 werden die Elemente der Nachrichten mit ihren Unterelementen dargestellt.

Die Abbildung 4.5 stellt die in der WSDL Datei definierten und bei den Elementen verwendeten Datentypen dar.

4.4.2 Modulübersicht

In diesem Abschnitt werden die einzelnen Module beschrieben:

- **openclone.webservice**
  - **main** beinhaltet den Servercode (Hintergrunddienst, Signalbehandlung) und startet den standalone Webserver des Webservices
  - **config** enthält die Standardkonfiguration
- **openclone.webservice.database**
  - **database** bietet eine vereinfachte, für das Webservice optimierte Schnittstelle zu SQLAlchemy

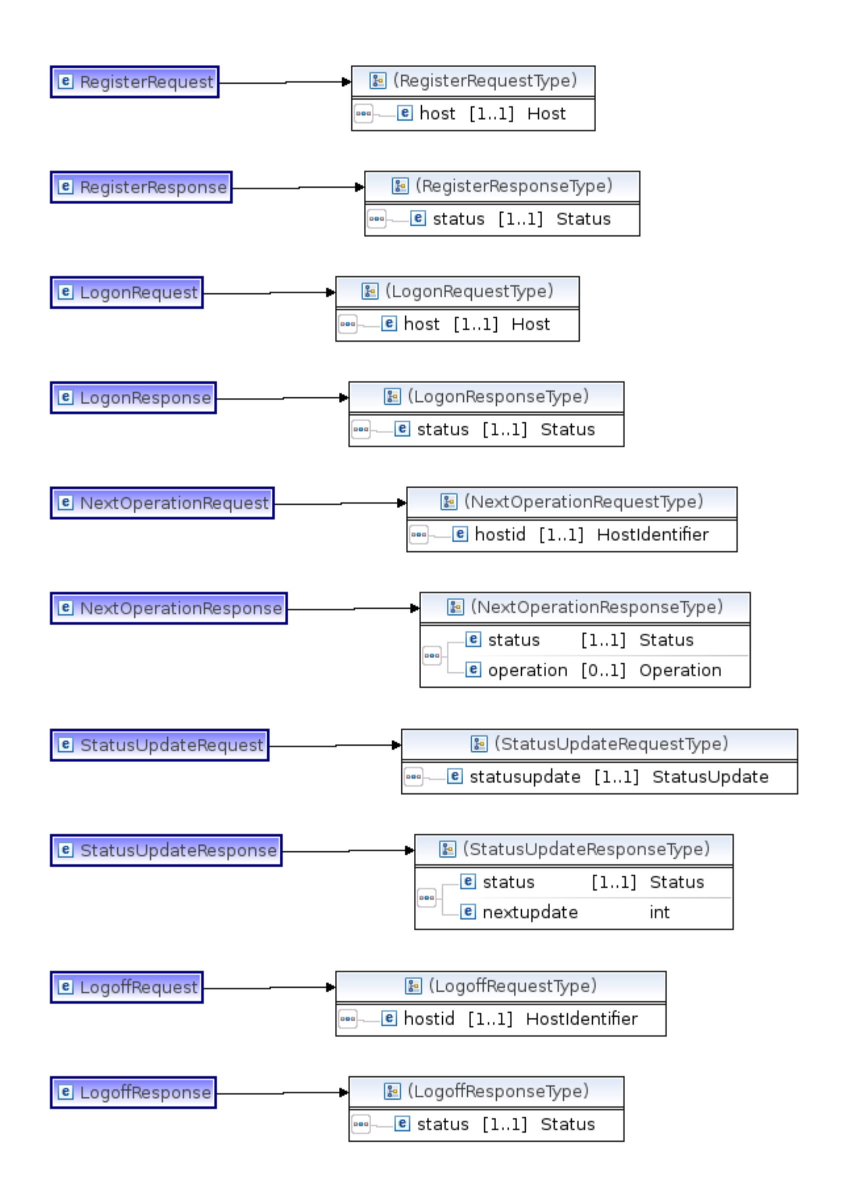


Abbildung 4.4: Webservice Elemente

- **tablemapping** beinhaltet die Klassendefinitionen der Tabellen

- **openclone.webservice.webservice**
  - **webservice** beinhaltet die Klasse zum Steuern des Webservers
  - **engineservice** beinhaltet die Implementierung der Webservice Methoden (von OpenCloneEngineService_services_server abgeleitet)
  - **OpenCloneEngineService_services_server** stellt die Schnittstelle zur Implementierung der Webservice Methoden dar (Automatisch generiert; siehe 4.4.4)
  - **OpenCloneEngineService_services_types** stellt die vom Webservice eingesetzten komplexen Datentypen als Klassen zur Verfügung (Automatisch generiert; siehe 4.4.4)

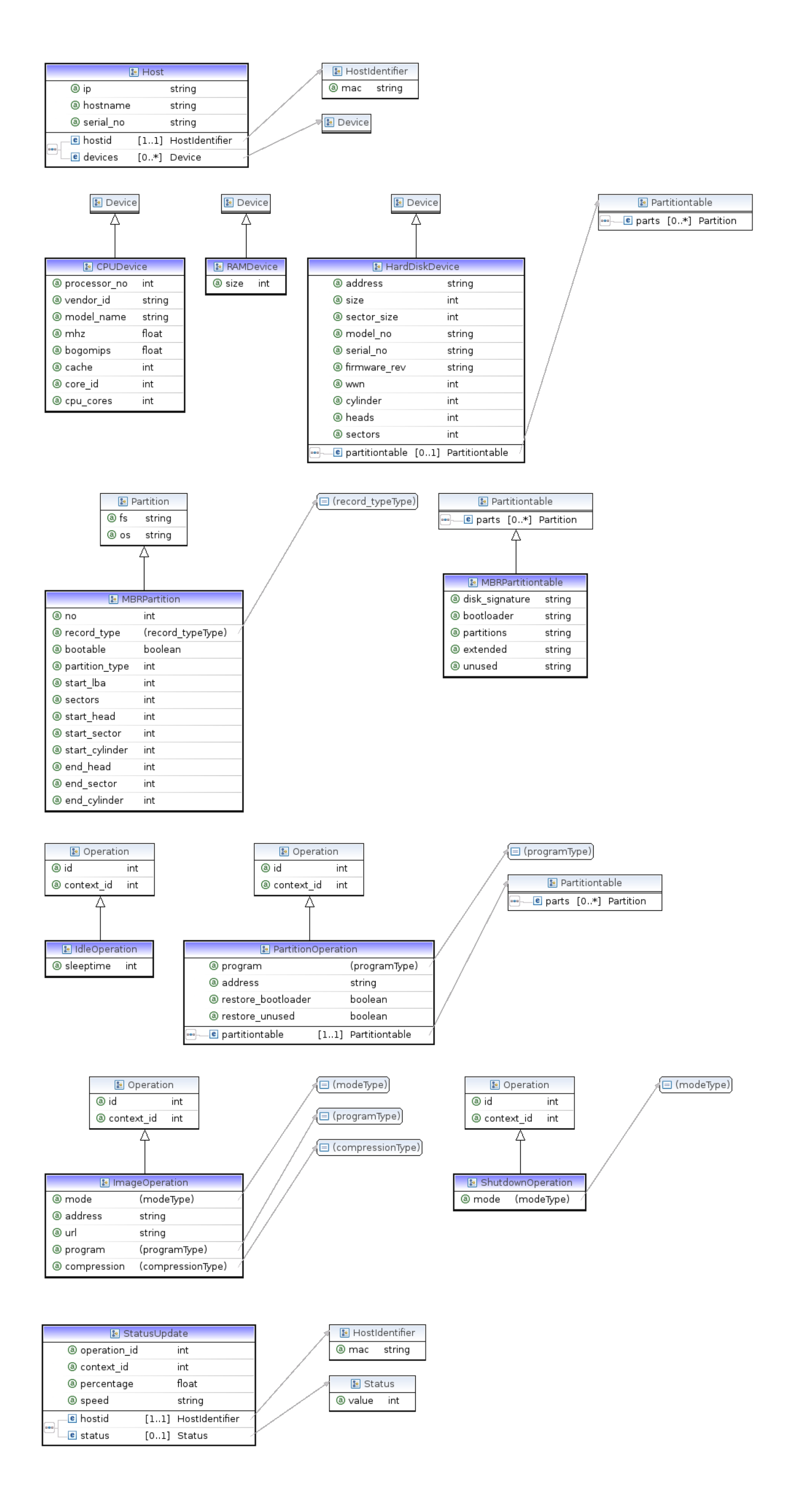


Abbildung 4.5: Webservice Datentypen

  - **OpenCloneEngineService_services** stellt die Methoden zum Aufruf des Werbservices zur Verfügung (Automatisch generiert; siehe 4.4.4)

- **openclone.common**
  - **nettools** stellt eine Netzwerktools zur Verfügung (z.B. Routing Table lookup)
  - **utils** stellt einige Hilfsmittel bei der Kommunikation mit Prozessen zur Verfügung
  - **version** beinhaltet die Programmversion

- **openclone.common.transfer** beinhaltet alle implementierten Übertragungsmethoden (siehe 4.3.5)
  - **udpcast** stellt eine Schnittstelle und Verwaltungstools für das Übertragungsprogramm UDPcast zur Verfügung

4.4.3 Abhängigkeiten

Für den Betrieb des OpenClone Webservices müssen folgende Abhängigkeiten erfüllt werden:

- Python 2.5 (python)
- Zolera SOAP Infrastructure 2.0 (python-zsi)
- SQLAlchemy >= 0.4.4
- MySQL für Python (python-mysqldb)
- UDPcast
- ip (iproute2)

4.4.4 SOAP Implementierung - Zolera SOAP Infrastructure (ZSI)[25]

Die Zolera SOAP Infrastructure stellt sowohl zum Erstellen von Webservice Clients als auch zum Erstellen von Webservice Servern je ein Utility zur Verfügung. Beide Utilities benötigen die WSDL Datei des Webservices als Eingabedatei und generieren einige Python Module als Ausgabe.

Webservice Client

Zum Erzeugen der für den Client benötigten Module dient der Befehl `wsdl2py`.

```
1 wsdl2py --file OpenCloneEngineService.wsdl --complexType --extended
```

Listing 4.2: Erstellen der benötigten Client Module

[25] [BM09, SB09]

Das Listing 4.2 zeigt ein Beispiel zum Erzeugen der Client Module für das OpenClone Webservice. Der Paramter `--complexType` aktiviert die Unterstützung für komplexe Datentypen, wie sie in der WSDL Datei verwendet werden. Der Parameter `--extended` aktiviert die Erzeugung von vereinfachten Webservice Methoden, welche nicht die gesamte Nachricht, sondern nur die Unterelemente des Nachrichtenelements als Parameter erwarten und die Nachricht selbst anlegen. Dadurch wird trotz der Limitierung auf eine einteilige Nachricht, wie bereits im Kapitel 4.4.1 erwähnt, ein direkter Aufruf einer Webservice Methode mit mehreren Parametern ermöglicht.

Unter den erzeugten Modulen befindet sich eines mit dem Namen `OpenCloneEngineService_services`, welches die Proxy Klasse `OpenCloneEngineServiceSOAPSOAP` zum Aufrufen des Webservices enthält. Dabei wird der Aufruf so vereinfacht, dass er sich kaum von einem lokalen Funktionsaufruf unterscheidet. Im Listing 4.3 wird ein einfaches Beispiel zum Aufruf einer Webservice Methode gegeben. Dabei wird die Methode `logon` des Webservices `http://localhost:18080/OpenCloneEngineService/` mit dem Host-Element als Parameter aufgerufen.

```
1  from OpenCloneEngineService_services import *
2
3  # Anlegen des Proxy Objekts fuer den Webserviceaufruf
4  service = OpenCloneEngineServiceSOAPSOAP(url='http://localhost:18080/
       OpenCloneEngineService/')
5  # Anlegen eines neuen host-Elements
6  host = logonRequest().new_host()
7  # Anlegen eines neuen hostid-Elements
8  host.Hostid = host.new_hostid()
9  # Setzen des mac-Attributs
10 host.Hostid.set_attribute_mac('000000000000')
11 # Aufrufen der Methode
12 status = service.logon(host)
```

Listing 4.3: Aufruf einer Webservice Methode mit ZSI in Python

Webservice Server

Zum Erzeugen der vom Server benötigten Module wird zusätzlich der Befehl `wsdl2dispatch` verwendet.

```
1 wsdl2py --file OpenCloneEngineService.wsdl --complexType --extended
2 wsdl2dispatch --file OpenCloneEngineService.wsdl --extended
```

Listing 4.4: Erstellen der benötigten Client Module

Das Listing 4.4 zeigt ein Beispiel zum Erzeugen der Server Module für das OpenClone Webservice. Die Parameter von `wsdl2py` wurden bereits beim Webservice Client besprochen, die Parameter des `wsdl2dispatch` Befehls sind sehr ähnlich. Der Parameter `--extended` bedeutet, wie bei `wsdl2py`, dass einfache Methoden verwendet werden können, welche nicht die komplette Nachricht, sondern nur die Unterelemente des Nachrichtenelements erwartet.

Das Listing 4.5 zeigt eine Implementierung einer Methode des OpenClone Webservices und den Start des Webservices.

```
1  from OpenCloneEngineService_services import *
2  from OpenCloneEngineService_services_server import OpenCloneEngineService
3  from ZSI.ServiceContainer import ServiceContainer
4
5  class OpenCloneEngineServiceImpl(OpenCloneEngineService):
6      def __init__(self):
7          OpenCloneEngineService.__init__(self, impl=self)
8
9      # Implementierung der logon Methode
10     def logon(self, host):
11         # Status Element anlegen
12         status = logonResponse().new_status()
13         # value Attribut setzen
14         status.set_attribute_value(0)
15         return status
16
17 # Server einrichten
18 servicecontainer = ServiceContainer(('', 18080))
19 # OpenClone Webservice hinzufuegen
20 servicecontainer.setNode(OpenCloneEngineServiceImpl())
21 # Starten des Servers
22 servicecontainer.serve_forever()
```

Listing 4.5: Implementierung eines Webservice Servers mit ZSI in Python

Probleme mit ZSI

Während der Entwicklung des Webservices sind einige Probleme mit ZSI aufgetreten, welche durch die eher kurz gehaltene Dokumentation schwer zu lösen waren. Oft war das Debuggen des ZSI Codes erforderlich, um die genaue Fehlerursache feststellen zu können. Dazu wurde der Python Debugger Winpdb[26] in der Version 1.3.8 eingesetzt.

Ein anfängliches Problem war auch, dass das Webservice im `rpc` Style angelegt wurde, welcher für entfernte Funktionsaufrufe ausgelegt ist. ZSI unterstützt jedoch den `rpc` Style nicht im `extended` Modus, welcher das Schreiben von weniger SOAP spezifischen Methoden erlaubt. Eine Diskussion auf der Python Web Services Mailinglist bestätigt dies. [Bov09] Deshalb wurde das Webservice auf den `document` Style umgestellt.

Der `simple-naming` Modus verspricht einfachere Benennung, jedoch ist er, wie in der Dokumentation beschrieben [BM09, S. 13], nicht mit der `complexType` Option kombinierbar. Die `complexType` Option ist allerdings unbedingt notwendig, weil sonst keine Generalisierung bei den Datentypen unterstützt wird. Jedoch ist das Instanziieren eines Kind-Datentyps ohne `simple-naming` kompliziert und zusätzliche Kenntnisse über den inneren Aufbau der für Client und Server generierten Module notwendig.

Weiters sind bei ZSI keine eigenen Fehler-Nachrichten möglich, denn es werden alle Fehler über die Exception ZSI.FaultException übertragen. Dabei wird dem Client ein Ablaufverfolgung (Trace) des

[26] Frei verfügbar unter `http://winpdb.org/`

serverseitig laufenden Codes als String mitgesendet.

Ein weiteres Problem stellt in der aktuellen Version 2.0 von ZSI der Datentyp Base64String dar. Normalerweise sollte die Daten eines Base64String Attributs automatisch bei der Übertragung mit Base64 kodiert und beim Empfang wieder dekodiert werden, ohne den Programmierer des Webservices damit zu belasten. Jedoch tritt im Kodierungsprozess ein Fehler auf, sodass ein späteres das Dekodieren nicht mehr möglich ist. Eine Analyse mit dem Netzwerksniffer Wireshark[27], hat ergeben, dass `&#xAU/VT9Q ==&#xA` anstatt des erwarteten Wertes `U/VT9Q==` übertragen wird. Die zusätzlichen, in HTML kodierten Anführungszeichen `&#xA` werden vom Empfänger natürlich bereits zum Base64String gerechnet, welcher die in Base64 unbekannten Zeichen nicht dekodieren kann. Als Abhilfe wurden alle im Webservice als Base64String definierten Attribute durch den String Datentyp ersetzt und die Kodierungs- und Dekodierungsarbeit selbst erledigt.

4.4.5 Objektrelationaler Mapper - SQLAlchemy[28]

SQLAlchemy ist ein objektrelationaler Mapper für Python. Das bedeutet er stellt die in der Datenbank verfügbaren Tabellen in Python als Objekte zur Verfügung und ermöglicht somit einen einfachen Zugriff auf die Daten. (siehe 3.8)

Tablemapping

Um SQLAlchemy mit allen Fähigkeiten einsetzen zu können, muss ein Tablemapping erstellt werden, welches alle benötigten Tabellen auf Klassen abbildet. Um das Tablemapping möglichst einfach zu gestalten, wird die Erweiterung `declarative` verwendet. Das Listing 4.6 zeigt einen Teil des OpenClone Tablemappings.

```
1  from sqlalchemy import *
2  from sqlalchemy.ext.declarative import declarative_base
3  from sqlalchemy.orm import relation, backref
4
5  Base = declarative_base()
6
7  class Group(Base):
8      __tablename__ = 'group'
9      id = Column('id', Integer, primary_key=True)
10     name = Column('name', String(255))
11
12 class Host(Base):
13     __tablename__ = 'host'
14     id = Column('id', Integer, primary_key=True)
15     group_id = Column('group_id', Integer, ForeignKey('group.id'))
16     group = relation('Group', backref='hosts')
17     mac = Column('mac', String(12), nullable=False)
18     ip = Column('ip', String(15))
```

[27]Frei verfügbar unter `http://www.wireshark.org/`
[28][SQL09]

```
19      hostname = Column('hostname', String(255))
20      serial_no = Column('serial_no', String(255))
21      running = Column('running', Boolean, nullable=False)
22      last_boot_time = Column('last_boot_time', DateTime)
23
24  class Device(Base):
25      __tablename__ = 'device'
26      id = Column('id', Integer, primary_key=True)
27      host_id = Column('host_id', Integer, ForeignKey('host.id'), nullable=
            False)
28      host = relation('Host', backref='devices')
29      type = Column('type', String(8), nullable=False)
30      __mapper_args__ = {'polymorphic_on': type}
31
32  class RAMDevice(Device):
33      __tablename__ = 'device_ram'
34      __mapper_args__ = {'polymorphic_identity': 'ram'}
35      id = Column('id', Integer, ForeignKey('device.id'), primary_key=True)
36      size = Column('size')
```

Listing 4.6: Ausschnitt aus dem OpenClone Tablemapping (tablemapping.py)

Die Tabelle `group` wird dabei auf die Klasse `Group` abgebildet. Sie beinhaltet den Primärschlüssel `id` von Typ Integer und die Spalte `name` vom Typ String. In der Klasse `Host` verweist der Fremdschlüssel `group_id` auf die Spalte `id` in der Tabelle `group`. Weiters wird mit der `relation` Funktion das Objekt der Gruppe unter dem Attribut `group` verfügbar gemacht und in der Klasse `Group` eine Liste namens `hosts` eingefügt, welche alle Hosts einer Gruppe beinhaltet. Die Klasse `Device` dient als Basisklasse für die `RAMDevice` Klasse und dabei wird in der Basisklasse in der Spalte `type` der speziellere Typ festgelegt. Um beim Auslesen eines Devices automatisch die spezialisierte Klasse, wie z.B. RAMDevice, zu erhalten, ist der Parameter `polymorphic_on` in der Basisklasse und `polymorphic_identiy` in der Subklasse notwendig. Dadurch wird das Auslesen sehr erleichtert, da mit einer einzigen Abfrage beispielsweise sofort ein RAMDevice Objekt verfügbar ist, ohne dass vorher bekannt sein muss, dass das Device vom Typ `ram` ist.

Datenbankzugriff mit SQLAlchemy

Im Listing 4.7 wird der Einsatz von SQLAlchemy gezeigt, dabei wird eine Verbindung aufgebaut, ein neuer Host und ein RAMDevice angelegt, der Host wieder ausgelesen, alle RAMDevices ausgegeben und gelöscht und die Verbindung wieder abgebaut.

```
1  from sqlalchemy import create_engine, sql
2  from sqlalchemy.orm import sessionmaker
3  from tablemapping import *
4
5  # Verbindung aufbauen
6  engine = create_engine('mysql://root:@localhost/openclone')
7  Session = sessionmaker(bind=engine)
```

```
8  session = Session()
9
10 # Neuen Host anlegen
11 host1 = Host(mac='000000000000', running=True, last_boot_time=sql.func.now()
       )
12
13 # Werte veraendern
14 host1.ip = '192.168.15.10'
15
16 # Device anlegen und dem Host zuordnen
17 device = RAMDevice(size=1024)
18 host1.devices.append(device)
19
20 # Host speichern
21 session.save(host1)
22
23 # Host mit MAC-Adresse 000000000000 auslesen
24 host2 = session.query(Host).filter_by(mac='000000000000').one()
25
26 # RAMDevices anzeigen und loeschen
27 for dev in host2.devices:
28     if type(dev) == RAMDevice:
29         print dev.size
30         session.delete(dev)
31
32 # Verbindung beenden
33 session.close()
```

Listing 4.7: Einsatz von SQLAlchemy

Mehrteilige Fremdschlüssel

Etwas komplizierter ist die Definition von mehrteiligen Fremdschlüsseln im Tablemapping, doch auch diese Aufgabe konnte nach einigen Anfangsschwierigkeiten mit Hilfe der guten Dokumentation [SQL09] gelöst werden. Ein Beispiel hierfür ist in Listing 4.8 zu sehen. Es ist notwendig das Tabellenobjekt selbst anzulegen, wobei die mehrteiligen Fremdschlüssel mittels `ForeignKeyConstraint` definiert werden. Es ist notwendig, dass `cur_operation` und `next_operation` erst nach der Klassendefinition angelegt werden, da auch die HostTask Klasse selbst voll verfügbar sein muss. Außerdem ist es bei diesen Beiden notwendig die Join Bedingung anzugeben, da diese durch ihre ähnlichen Fremdschlüssel nicht automatisch ermittelt werden kann.

```
1 class HostTask(Base):
2     __table__ = Table('host_task', Base.metadata,
3         Column('host_id', Integer, ForeignKey('host.id'), primary_key=True),
4         Column('task_id', Integer, ForeignKey('task.id'), primary_key=True),
5         Column('status', String(8), nullable=False),
6         Column('cur_operation_id', Integer),
```

```
7           Column('next_operation_id', Integer),
8           Column('percentage', Float),
9           Column('speed', String(255)),
10          ForeignKeyConstraint(['task_id', 'cur_operation_id'],
11                               ['operation.task_id', 'operation.id']),
12          ForeignKeyConstraint(['task_id', 'next_operation_id'],
13                               ['operation.task_id', 'operation.id'])
14      )
15      host = relation('Host', backref='hosttasks')
16      task = relation('Task', backref='taskhosts')
17
18  HostTask.cur_operation = relation('Operation', primaryjoin=and_(HostTask.
        task_id==Operation.task_id, HostTask.cur_operation_id==Operation.id))
19  HostTask.next_operation = relation('Operation', primaryjoin=and_(HostTask.
        task_id==Operation.task_id, HostTask.next_operation_id==Operation.id))
```

Listing 4.8: Erweiterter Ausschnitt aus dem OpenClone Tablemapping (tablemapping.py)

4.5 Webinterface

Das OpenClone Webinterface stellt die Schnittstelle zwischen dem Benutzer und dem Cloningsystem dar. Es bietet dem Benutzer die Möglichkeit, das gesamte System über eine grafische Oberfläche zu steuern. Dabei muss keine zusätzliche Software auf dem Computer installiert werden, von dem aus die Steuerung erfolgt. Es genügt ein Webbrowser, welcher üblicherweise auf jedem Computer bereits installiert ist. Somit kann der Benutzer die Software von jedem Computer bedienen, von dem aus er auf den Server zugreifen kann, auf dem das Webinterface zur Verfügung steht.

Das OpenClone Webinterface ist in PHP realisiert. Dem entsprechend ist ein Webserver mit PHP Support zum Betrieb erforderlich. Entwickelt und getestet wurde die Software unter Ubuntu Linux mit Apache 2. Zum Zugriff auf die Datenbank wird ein objektrelationaler Mapper namens Doctrine verwendet. Dieser ermöglicht den Datenbankzugriff unter PHP, wobei der Programmierer auf einem hohen Abstraktionslevel arbeiten kann. Doctrine benötigt mindestens die PHP Version 5.2.3. Die Verwendung von Doctrine wird im Kapitel 4.5.3 noch genauer erklärt.

4.5.1 Menüstruktur

Nach erfolgreicher Anmeldung des Benutzers mittels Benutzername und Passwort am Webinterface erscheinen ihm folgende Menüpunkte zur Auswahl.

- Overview
- Image Creation Wizard
- Deployment Wizard
- Advanced

- Hostlist
- Grouplist
- Tasklist
- Imagelist
- Status
- Settings

In der Abbildung 4.6 ist die Menüstruktur grafisch veranschaulicht. Die Aufteilung der Seiten, die sich hinter den einzelnen Menüpunkten befinden, sowie eine Beschreibung dieser, wird im folgenden Abschnitt (4.5.2 Seitenaufteilung) erläutert.

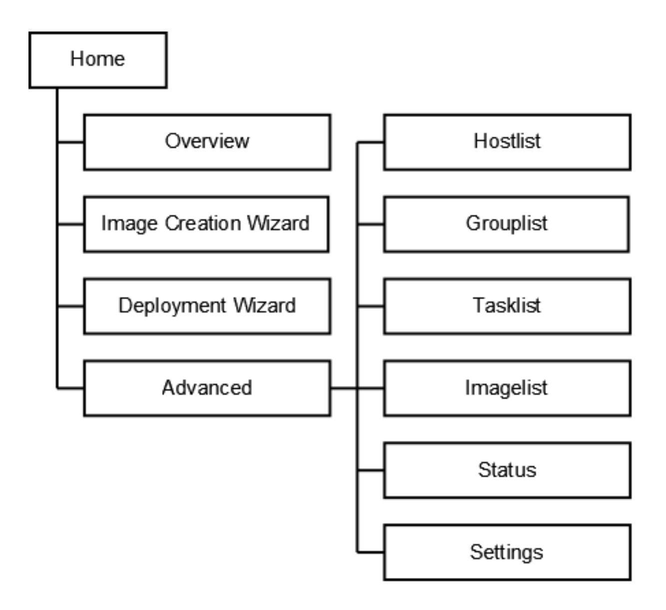


Abbildung 4.6: Menüstruktur des OpenClone Webinterface

4.5.2 Seitenaufteilung

Das OpenClone Webinterface besteht aus mehreren PHP Seiten und zugehörigen PHP Dateien mit Funktionen, die der Seite zur Verfügung gestellt werden. Die Datei mit den Funktionen hat den gleichen Namen wie die anzuzeigende Seite selber, jedoch mit dem Postfix '_func'. So existiert zum Beispiel eine Seite 'hostlist.php', welche die Datei 'hostlist_func.php' einbindet. Diese Datei enthält alle relevanten Funktionen, die in der Datei 'hostlist.php' verwendet werden. Es wurde versucht, möglichst wenig von der Programmlogik direkt in die Dateien zu implementieren, welche vom Webbrowser aufgerufen werden. Stattdessen werden dort, wenn möglich, fast nur Funktionen der eingebundenen Funktionsdateien verwendet. Dies soll die Übersichtlichkeit und Strukturierung des Quellcodes fördern. Weiters befinden sich die anzuzeigenden Seiten und die Funktionsdateien jeweils in einem separaten Verzeichnis.

Im Folgenden werden die Funktionen der einzelnen Seiten und der dazugehörigen Funktionsseiten kurz beschrieben.

Allgemeine Seiten

Dieser Abschnitt behandelt allgemeine Seiten, die keinem Wizard zugeordnet sind oder unter dem Menüpunkt 'Advanced' zu finden sind.

- index.php Dies ist die Startseite des Webinterface. Sie wird bei jedem Request eines Webbrowser aufgerufen und stellt das gesamte Design dar. Weiters bindet sie die jeweils angeforderte Inhaltsseite ein. Es sind Funktionen zur Benutzerauthentifizierung (also zum Anmelden) und ebenfalls zum Abmelden implementiert.

- overview.php Diese Seite zeigt Informationen über den Computer an, auf dem das Webinterface installiert ist. Der Benutzer hat so einen schnellen Überblick über die Auslastung des Rechners sowie der Festplatten. Weiters wird die Versionsnummer der OpenClone Webinterface Software angezeigt.

Image Creation Wizard

Der Image Creation Wizard hat die Aufgabe, dem Benutzer möglichst einfach das Anlegen eines Festplattenabbildes eines Computers zu ermöglichen. Dabei wird der Wizard in 4 Schritte unterteilt.

- create1.php
  Diese Seite listet alle verfügbaren, zur Zeit eingeschalteten Clients auf, unter denen der Benutzer jenen wählen kann, von dem das Festplattenabbild erzeugt werden soll.

- create2.php
  Auf dieser Seite werden die Festplatten des gewählten Computers gelistet. Der Benutzer wählt hier die Festplatte, von der das Image erstellt werden soll.

- create3.php
  Es werden zusammenfassend Details über den gewählten Client und die gewählte Festplatte angezeigt. Der Benutzer vergibt nun noch einen Namen für das Abbild, wählt eine Kompressionsmethode und kann bei Bedarf auch noch auswählen, ob der Client nach beendetem Clonevorgang heruntergefahren oder neu gestartet werden soll.

- create4.php
  Abschließend werden alle für die Abbilderzeugung relevanten Informationen in die Datenbank eingetragen und dem Benutzer wird die Möglichkeit geboten, auf eine Statusseite zu wechseln, um den Fortschritt des Klonvorganges zu verfolgen.

Deployment Wizard

Der Deployment Wizard ermöglicht dem Benutzer, auf einfache Weise, ein angelegtes Abbild einer Festplatte auf einen oder mehrere Rechner gleichzeitig zu verteilen. Dieser Wizard besteht, wie der Image Creation Wizard ebenfalls aus 4 Schritten.

- deploy1.php
  Zuerst werden alle am OpenClone Server verfügbaren Images angezeigt. Aus diesen wählt der Benutzer das gewünschte Image aus, welches verteilt werden soll.

- deploy2.php
  Auf dieser Seite werden alle eingeschalteten Clients aufgelistet. Der Benutzer kann einen einzlnen Client oder eine Gruppe von Clients wählen. Ist letzteres der Fall, werden alle Computer, die der Gruppe angehören, angezeigt. Nun werden die gewünschten Clients ausgewählt, auf welche das zuvor gewählte Abbild übertragen werden soll.

- deploy3.php
  Diese Seite zeigt die Festplatten an, auf die das gewählte Image übertragen werden kann. Dabei werden nur Festplatten dargestellt, welche in jedem Client vorhanden sind und die selbe Geräteadresse im Clientsystem besitzen. Ist in jedem gewähltem Client zum Beispiel eine Festplatte unter der Adresse '/dev/sda' verfügbar, wird sie angezeigt. Zusätzlich wird noch die kleinste Plattengröße aus dieser Festplattengruppe angezeigt. Ist diese Größe kleiner als die Größe des zu übertragenden Images, kann diese Festplatte vom Benutzer nicht gewählt werden. Aus allen auswählbaren Festplatten wird eine vom Benutzer selektiert. Das Abbild wird dann auf jedem Client auf die Festplatte gespeichert, welche die selbe Geräteadresse hat, die auch die gewählte Festplatte hat (zum Beispiel '/dev/sda'). Bei Bedarf kann auch noch veranlasst werden, dass die Clients nach der Imageübertragung heruntergefahren oder neu gestartet werden sollen.

- deploy4.php
  Zum Schluss werden die benötigten Informationen zur Wiederherstellung eines Abbildes auf einen oder mehrere Computer in die Datenbank eingetragen. Es besteht für den Benutzer ebenfalls hier die Möglichkeit, auf die Statusseite zu wechseln, um den Fortschritt der Wiederherstellung zu sehen.

Advanced

Unter dem Menüpunkt Advanced sind folgende Seiten zusammengefasst, welche erweiterte Einstellungen und Operationen des Cloningsystems zur Verfügung stellen.

- hostlist.php
  Diese Seite listet im Cloningsystem registrierte Clients auf. Dabei kann ausgewählt werden, ob nur eingeschaltete Clients, oder nur Clients einer bestimmten Gruppe, oder auch alle Clients angezeigt werden. Der Benutzer kann hier folgende Einstellungen vornehmen.
  - Eine Abfolge von Befehlen (einen Task) einem oder mehreren Hosts zuweisen
  - Den Hostnamen eines oder mehrerer Clients gleichzeitig ändern
  - Die IP Adresse eines oder mehrerer Clients gleichzeitig ändern
  - Einen oder mehrere Hosts einer Hostgruppe zuordnen

- hostdetails.php
  Auf dieser Seite werden folgende detailierte Informationen zu dem gewählten Client angezeigt

  - Generelle Informationen (Hostname, IP Adresse, MAC Adresse, Seriennummer, Hostgruppe, letzte Anmeldung am Cloningsystem)
  - Prozessoren (Hersteller, Modell, Taktfrequenz)
  - RAM (Größe)
  - Festplatten (Geräteadresse, Größe, Modellnummer, Seriennummer)
  - Tasks, die dem Client zugeordnet sind

  Weiters kann der Benutzer folgende Informationen des Computers ändern.

  - Hostname
  - IP Adresse
  - MAC Adresse
  - Seriennummer

- grouplist.php
  Diese Seite listet alle verfügbaren Hostgruppen auf und bietet die Möglichkeit, neue Gruppen zu erstellen sowie bereits vorhandene Gruppen wieder zu löschen.

- groupedit.php
  Auf dieser Seite werden alle Hosts, die einer gewählten Gruppe angehören, aufgelistet und können vom Benutzer aus dieser Gruppe entfernt werden.

- tasklist.php
  Alle angelegten Tasks werden auf dieser Seite angezeigt. Hier besteht die Möglichkeit, einen neuen Task anzulegen.

- taskedit.php
  Diese Seite zeigt alle Operationen an, die dem gewählten Task zugeordnet sind. Diese Operationen können vom Task entfernt werden. Weiters können neue Operationen zum Herunterfahren oder Neustarten eines Clients zum gewählten Task hinzugefügt werden.

- imagelist.php
  Hier werden alle verfügbaren Festplattenabbilder dargestellt. Diese Abbilder können vom Benutzer gelöscht werden. Weiters kann der Name der Images geändert werden.

- status.php
  Auf dieser Seite werden alle laufenden und bereits beendeten Tasks inklusive deren Fortschritt angezeigt. Damit hat der Benutzer eine Übersicht über den Status seiner angelegten Aufgaben.

- settings.php
  Diese Seite bietet die Möglichkeit, Benutzername und Passwort für die Authentifizierung zu ändern.

4.5.3 Objektrelationaler Mapper - Doctrine

Doctrine ist ein objektrelationaler Mapper für die Programmiersprache PHP[29]. Solche Mapper dienen dem Datenbankzugriff und erleichtern dabei dem Softwareentwickler die Arbeit. (siehe 3.8)

Doctrine ist zum Zeitpunkt der Anwendungswentwicklung in Version 1.1.0 verfügbar. Es bietet unter anderem folgende Möglichkeiten, welche auch bei der Entwicklung des OpenClone Webinterfaces genutzt wurden. [Doc09]

- Automatische Generierung der PHP Klassen für Tabellen aus einer bereits vorhandenen Datenbank
- Eine Abfragesprache für SQL namens DQL (Doctrine Query Language), welche an die HQL (Hibernate Query Language) des objektrelationalen Mappers Hibernate für Java angelehnt ist
- Beziehungsauflösung zwischen Tabellen einer Datenbank
- Generalisierung von Tabellen

Klassenaufbau

Doctrine arbeitet für jede verwendete Tabelle der Datenbank mit einer eigenen PHP Klasse, welche die benötigten Attribute und Methoden enthält. Jede Klasse, die eine Relation darstellt, muss von der Basisklasse 'Doctrine_Record' abgeleitet werden. In der Basisklasse werden Methoden definiert, die zur Beschreibung der Relation und zur Durchführung unterschiedlichster Operationen auf diese dienen. Im Listing 4.9 wird die Klasse der Relation 'user' dargestellt. Die Funktion 'setTableDefinition' dient zur Beschreibung der Relation und der Attribute. Mit `$this->setTableName('user')` wird der Tabellenname auf 'user' festgelegt. Mit `$this->hasColumn` werden die Attribute definiert. Die Tabelle 'user' hat folgende Attribute.

- id (Integer, autoincrement)
- name (varchar, not null)
- password (varchar, fixe Länge, not null)
- last_login (timestamp)

In der Funktion 'setUp' werden Beziehungen dargestellt. Da die Tabelle 'user' keine Beziehung hat, ist diese Funktion in dieser Klasse auch nicht vorhanden.

```
1 class User extends Doctrine_Record
2 {
3   public function setTableDefinition()
4   {
5     $this->setTableName('user');
```

[29] Frei verfügbar unter http://www.doctrine-project.org/

```
6      $this->hasColumn('id', 'integer', 4, array('type' => 'integer', 'length'
           => 4, 'unsigned' => 1, 'primary' => true, 'autoincrement' => true));
7      $this->hasColumn('name', 'string', 255, array('type' => 'string', '
          length' => 255, 'default' => '', 'notnull' => true));
8      $this->hasColumn('password', 'string', 40, array('type' => 'string', '
          length' => 40, 'fixed' => true, 'default' => '', 'notnull' => true));
9      $this->hasColumn('last_login', 'timestamp', null, array('type' => '
          timestamp'));
10   }
11 }
```

Listing 4.9: Doctrine 'user' Klasse

Generalisierung

Zur Veranschaulichung der Generalisierung werden die beiden Tabellen 'operation' und 'operation_-shutdown' aus der Datenbank verwendet (siehe Abbildung 4.1). Die Relation 'operation' ist die allgemeine Relation und beinhaltet Attribute, welche jede der speziellen Relationen enthalten soll. Für die Generalisierung wichtig ist das Attribut 'type' vom Datentyp ENUM. Der Wert dieses Attributes gibt an, welche spezielle Relation zu einem Tupel in der generellen Relation gehört. Angenommen ein Tupel in der Relation 'operation' hat für 'type' den Wert 'shutdown' eingetragen, dann wird die 'operation_shutdown' Relation benötigt, um die speziellen Attribute zu verwenden. Um das Tupel der generellen Relation eindeutig mit dem richtigen Tupel der speziellen Relation zu verknüpfen, wird eine 1:1 Beziehung der beiden 'id' Attribute verwendet, welche für die jeweilige Relation den Schlüssel darstellen.

Die beiden Klassen 'operation' und 'operation_shutdown' werden, wie in Listing 4.10 und Listing 4.11 veranschaulicht, in PHP Klassen dargestellt.

```
1  class Operation extends Doctrine_Record
2  {
3    public function setTableDefinition()
4    {
5      $this->setTableName('operation');
6      $this->hasColumn('id', 'integer', 4, array('type' => 'integer', 'length'
           => 4, 'unsigned' => 1, 'primary' => true, 'autoincrement' => true));
7      $this->hasColumn('task_id', 'integer', 4, array('type' => 'integer', '
          length' => 4, 'unsigned' => 1, 'default' => '', 'notnull' => true));
8      $this->hasColumn('sequence', 'integer', 4, array('type' => 'integer', '
          length' => 4, 'unsigned' => 1, 'default' => '', 'notnull' => true));
9      $this->hasColumn('type', 'enum', 9, array('type' => 'enum', 'length' =>
          9, 'values' => array(0 => 'partition', 1 => 'image', 2 => 'shutdown')
          , 'default' => '', 'notnull' => true));
10
11     $this->setSubClasses(array('OperationImage'    => array('type' => 'image
          '),
```

```
12                              'OperationPartition' => array('type' => '
                                   partition'),
13                              'OperationShutdown' => array('type' => '
                                   shutdown')));
14   }
15
16   public function setUp()
17   {
18     $this->hasOne('Task', array('local' => 'task_id',
19                                'foreign' => 'id'));
20   }
21 }
```

Listing 4.10: Doctrine 'Operation' Klasse

```
1  class OperationShutdown extends Operation
2  {
3    public function setTableDefinition()
4    {
5      $this->setTableName('operation_shutdown');
6      $this->hasColumn('id', 'integer', 4, array('type' => 'integer', 'length'
          => 4, 'unsigned' => 1, 'primary' => true));
7      $this->hasColumn('mode', 'enum', 8, array('type' => 'enum', 'length' =>
         8, 'values' => array(0 => 'poweroff', 1 => 'reboot', 2 => 'exit'), '
         default' => '', 'notnull' => true));
8    }
9
10   public function setUp()
11   {
12     $this->hasOne('Operation', array('local' => 'id',
13                                     'foreign' => 'id'));
14   }
15 }
```

Listing 4.11: Doctrine 'OperationShutdown' Klasse

Zu beachten ist, dass die Klasse der speziellen Relation von der Klasse, welche die generelle Relation implementiert, abgeleitet werden muss. In der Funktion 'setTableDefinition' der Klasse 'operation' wird mittels `$this->setSubClasses` festgelegt, welche speziellen Klassen existieren und welchem 'type' Wert sie zugewiesen werden. In der Klasse 'operation_shutdown' wird in der Funktion 'setUp' der Fremdschlüssel mit `$this->hasOne` definiert.

Will man nun ein Tupel in die Relation 'shutdown_operation' einfügen, so muss man nur eine neue Instanz der Klasse 'OperationShutdown' anlegen und die Attribute mit den gewünschten Werten belegen. Dabei kann auch gleich auf die Attribute der Basisklasse über die Instanz oder speziellen Klasse zugegriffen werden. Dabei ist zu beachten, dass man unter Doctrine das Attribute 'type' selbst mit dem entsprechenden Wert belegen muss. Andere objektrelationale Mapper kümmern sich automatisch darum. Mit dem Statement `$this->save()` wird das Tupel in die Tabelle eingefügt. Doctrine kümmert

sich automatisch um die Aufteilung der Werte in ein jeweiliges Tupel der generellen und der speziellen Tabelle in der Datenbank. Im Listing 4.12 wird das Einfügen eines Tupels in die Tabelle 'operation_shutdown' gezeigt.

```
1 OperationShutdown $shutdown = new OperationShutdown();
2 //$shutdown->id = ... wird automatisch gesetzt
3 $shutdown->task_id = 2;
4 $shutdown->sequence = 1;
5 $shutdown->type = 'shutdown';
6 $shutdown->mode = 'poweroff';
7 $shutdown->save();
```

Listing 4.12: Einfügen eines Tupels in 'operation_shutdown'

Um ein Tupel aus der Relation 'operation' auszulesen und dabei auch gleich die Attribute der speziellen Relation zu erhalten, würde in anderen objektrelationalen Mappern mit dem Statement `Operation $operation = Doctrine_Query::create()->from(Operation)`. Diese Abfrage würde ein Array von Objekten der Klasse 'Operation' liefern, über welche man auch auf die Attribute der speziellen Klasse, zugreifen kann. Die spezielle Klasse und das Objekt davon werden normalerweise vom Mapper selbst nach dem Wert 'type' Feld des Objektes von der Klasse 'Operation' ausgewählt. Unter Doctrine funktioniert diese Vorgehensweise jedoch leider nicht so einfach. Um an die Information eines Tupels einer speziellen Klasse zu gelangen, muss zuerst das Tupel der generellen Relation verarbeitet werden. Je nach Wert des Attributes 'type' muss dann ein Tupel aus der speziellen Relation ausgelesen werden. Bezogen auf die Relationen 'operation' und 'operation_shutdown' muss im Tupel der Relation 'operation' für 'type' der Wert 'shutdown' gesetzt sein. Danach kann das zugehörige Tupel aus der Relation 'operation_shutdown' abgefragt werden, indem eine Selektion des 'id' Attributes auf den Wert des 'id' Attributes des Tupels der 'operation' Relation angewendet wird. Danach erhält man die Attributwerte für die Attribute der speziellen Relation. Das Listing 4.13 veranschaulicht diese Vorgehensweise.

```
1 //Auslesen des generellen Tupels
2 $operation = Doctrine_Query::create()->from('Operation')->where('id=1')->
    execute();
3 //Hier kann auf die generellen Attribute zugegriffen werden
4 echo $operation[0]->type;
5
6 //Annahme: $operation->type = 'shutdown';
7 $shutdownOperation = Doctrine_Query::create()->from('OperationShutdown')->
    where('id='.$operation[0]->id)->execute();
8 //Hier kann auf die speziellen Attribute (im Fall von OperationShutdown:
    mode) zugegriffen werden
9 echo $shutdownOperation[0]->mode;
```

Listing 4.13: Auslesen eines Tupels aus der Relation 'operation' und der zugehörigen speziellen Relation 'operation_shutdown'

Löschen von Tupeln

Das Löschen eines Tupels einer Tabelle kann grundsätzlich sehr einfach mit der Doctrine Query Language erledigt werden. Im Listing 4.14 wird mit einer Doctrine Query Language Abfrage jenes Tupel aus der Tabelle 'user' (siehe Abbildung 4.1) gelöscht, welches '1' als Attributwert für das Attribut 'id' eingetragen hat.

```
1 Doctrine_Query::create()->delete()->from('User')->where('id=1')->execute();
```

Listing 4.14: Löschen eines Tupels mit der Doctrine Query Language

Bei Relationen, welche miteinander in Beziehung stehen, müssen zuerst alle Tupel aus der Relation entfernt oder der Wert des Attributes, welches den Fremdschlüssel definiert, geändert werden, in welchen auf den Schlüssel des zu löschenden Tupels der anderen Relation verwiesen wird, sonst kann man das Tupel aus der Relation, auf die verwiesen wird, nicht gelöscht werden. Zur Veranschaulichung werden die Relationen 'host' und 'group' aus der Datenkbank verwendet (siehe Abbildung 4.1). Angenommen, die Gruppe mit dem Wert '1' für das Attribut 'id' soll gelöscht werden. Dies ist jedoch nur möglich, wenn zuvor in jedem Tupel der Relation 'host', welches '1' als Wert für 'group_id' eingetragen hat (also in Beziehung mit der Gruppe mit der id '1' steht), geändert werden. Im Listing 4.15 ist genau dies dargestellt.

```
1  //Fremdschluessel aendern
2  $hosts = Doctrine_Query::create()->from('Host')->where('group_id=1')->
       execute();
3  foreach($hosts as $host)
4  {
5       $host->group_id = NULL;
6           $host->save();
7  }
8  //Alternative: Hosts loeschen
9  //Doctrine_Query::create()->delete()->from('Host')->where('group_id=1')->
       execute();
10
11 //Gruppe loeschen
12 Doctrine_Query::create()->delete()->from('Group')->where('id=1')->execute();
```

Listing 4.15: Löschen eines Tupels mit einer Beziehung

Soll nun solch ein Tupel einer Relation gelöscht werden, die mit einer anderen Relation in Beziehung steht und dabei die Attribute der Beziehung auch gleichzeitig Primärschlüssel für die jeweilige Relation sind, führt diese Vorgehensweise bei Doctrine zu einem internen JOIN Fehler. Abhilfe schafft hier, die Tupel der Relation, die den Fremdschlüssel definiert hat, nicht direkt zu löschen, sondern zuerst auszulesen und dann erst mit einem eigenen Statement zu löschen. Danach kann das Tupel der Relation gelöscht werden, auf die der Fremdschlüssel definiert war. Diese Vorgehensweise erfordert leider mehr Statements. In der Datenbank besteht zwischen den Tabellen 'partition' und 'image_partition' eine 1:1 Beziehung, wobei bei jeder Tabelle das Attribut 'id' für die Beziehung verwendet wird, welches auch

Primärschlüssel ist (siehe Abbildung 4.1). Um nun das Tupel der Tabelle 'partition' mit dem Attributwert '1' für das Attribut 'id' zu löschen, ist es erforderlich, auch aus der Tabelle 'image_partition' das Tupel mit dem Wert '1' für 'id' zu löschen. Da die beiden Attribute der Beziehung auch Primärschlüssel sind, muss das beschriebene Verfahren angewendet werden. Es wird das Tupel der 'image_partition' mit dem 'id' Wert '1' ausgelesen und mit einem eigenen Statement gelöscht. Danach kann das Tupel der Tabelle 'image' gelöscht werden. Dieser Vorgang wird im Listing 4.16 gezeigt.

```
1 //Tupel aus Tabelle 'image_partition' mit id=1 auslesen
2 $imagePartition = Doctrine_Query::create()->from('ImagePartition')->where('
    id=1')->execute();
3 //Tupel loeschen
4 $imagePartition[0]->delete();
5
6 //Tupel aus Tabelle 'partition' loeschen
7 Doctrine_Query::create()->delete()->from('Partition')->where('id=1')->
    execute();
```

Listing 4.16: Löschen eines Tupels mit einer Beziehung, bei der die Attribute der Beziehung gleichzeitig Primärschlüssel sind

4.5.4 Webinterface Wrapper

Die einzige Operation, bei der das Webinterface Zugriff auf das lokale Dateisystem der Serverfestplatte benötigt, ist jene zum Löschen bereits angelegter Festplattenabbilder. Diese Abbilder befinden sich in einem Verzeichnis, auf das jener Benutzer, unter dem der Apache Webserver Prozess läuft (im Fall von Ubuntu Linux: 'www-data'), keine Schreibrechte besitzt. Diese Maßnahme wurde aus Sicherheitsgründen getroffen. Es darf nur der Benutzer 'openclone-websrv' auf das Verzeichnis zugreifen, welcher das Image anlegen und wieder für Übertragungen zu Computern lesen darf. Weiters gehört das Verzeichnis der Gruppe 'openclone', welche ebenfalls Schreibrechte auf das Verzeichnis besitzt. Wenn nun ein Benutzer im Webinterface ein Festplattenabbild löscht, muss dieses auch auf der Festplatte des Servers gelöscht werden. Dazu muss der Systembenutzer 'www-data' die Möglichkeit haben, diesen Löschbefehl auszuführen, weil das Webinterface unter diesem Benutzer läuft. Dazu wurde ein Shell Script implementiert, welches ein Verzeichnis mit einem Image von der Festplatte löschen kann. Damit dieses Script nicht verändert werden kann, gehört es dem Benutzer 'root'. Es darf jedoch von jedem Benutzer ausgeführt werden. Wenn nun aber ein Benutzer dieses Script ausführt, der nicht zur Gruppe 'openclone' gehört, wie es auch beim Benutzer 'www-data' der Fall ist, kann er keine Daten vom Imageverzeichnis löschen. Um das Löschen möglich zu machen, darf der Benutzer 'www-data' dieses Wrapper Script als Benutzer 'openclone-webif' ausführen. Dieser Benutzer ist Mitglied in der Gruppe 'openclone' und hat die erforderlichen Rechte, um ein Image von der Platte zu entfernen. In Ubuntu Linux muss in der Datei '/etc/sudoers' folgende Zeile eingetragen werden, um es 'www-data' zu ermöglichen, das Wrapper Script als 'openclone-webif' auszuführen: `www-data ALL=(openclone-webif) NOPASSWD: /usr/bin/openclone-webinterface-wrapper`. Erst damit kann der Benutzer 'www-data' das Löschen eines Festplattenabbildes als Benutzer 'openclone-webif' durchführen.

4.6 Installer

Der Installer ist jene Software, welche sich um die Installation und Einrichtung der OpenClone Software sowie spätere Updates kümmert. Der Installer wurde als Shell Script implementiert, um möglichst wenige Abhängigkeiten zu benötigen und gleichzeitig Einschränkungen auf bestimmte Prozessor Architekturen zu umgehen. Die Installation der beiden OpenClone Softwarekomponenten Engine und Webservice wurden mit Hilfe des `setuputils`-Moduls in Python implementiert.

Der Installer ist zur Zeit nur für den Betrieb unter Ubuntu Linux 8.04 Hardy Heron mit Long-Term-Support ausgelegt. Dies soll jedoch später noch erweitert werden. (siehe 5.3.3) Weiters werden zur Zeit alle OpenClone Softwarekomponenten, also Engine, Webservice und Webinterface, vom Subversion Repository bezogen. Dies muss für spätere Installerversionen noch geändert werden, da in den meisten Fällen nicht die aktuelle Entwicklerversion, sondern die neueste stabile Version gewünscht ist.

4.6.1 Tätigkeiten des Installers

In diesem Abschnitt werden die Tätigkeiten erklärt, welche der Installer im Installationsmodus durchführt.

Abfragen der Benutzereinstellungen

Es werden die notwendigen Informationen zum Netzwerk und zum Image Speicherort abgefragt.

Ubuntu Systemupdate durchführen

Es wird mit Hilfe des Paketmanagement Tools `aptitude` oder `apt-get` ein Systemupdate durchgeführt.

Benötigte Software installieren

Folgende Softwarepakete werden benötigt:

- DHCP Server: ISC dhcpd[30] (dhcp3-server)
- TFTP Server: HPAs tftp-Server[31] (tftpd-hpa)
- NFS Server: NFS kernel server[32] (nfs-kernel-server)
- debootstrap[33]
- Bootloader: SYSLINUX[34] (syslinux)

[30] Frei verfügbar unter `http://www.isc.org/software/dhcp`
[31] Frei verfügbar unter `http://www.kernel.org/pub/software/network/tftp/`
[32] Frei verfügbar unter `http://nfs.sourceforge.net/`
[33] Frei verfügbar unter `http://packages.debian.org/stable/admin/debootstrap`
[34] Frei verfügbar unter `http://syslinux.zytor.com/`

- Versionsverwaltung: Subversion[35] (subversion)
- OpenClone Abhängigkeiten:
  - Python (python)
  - Zolera SOAP Infrastructure (python-zsi)
  - MySQL für Python (python-mysqldb)
  - SQLAlchemy (python-sqlalchemy)
  - UDPcast (udpcast)
  - Apache 2 (apache2)
  - PHP 5 (php5)
  - MySQL Modul für PHP 5 (php5-mysql)
  - MySQL Server (mysql-server)

DHCP Server konfigurieren

Der DHCP Server wird nach den zuvor abgefragten Einstellungen des Benutzers konfiguriert.

Bootloader (pxelinux) einrichten

Es wird eine Standardkonfiguration erstellt, welche später das auf den Clients installierte Ubuntu System starten soll.

TFTP Server konfigurieren

Es wird der Daten-Pfad des TFTP Servers eingerichtet. Zur Zeit ist dieser Schritt noch nicht automatisiert und es muss die Konfigurationsdatei `/etc/inetd.conf` noch händisch angepasst werden.

Root Dateisystem für Clients einrichten

Zur Installation des Ubuntu Systems für Clients werden folgende Schritte ausgeführt:

- Installation eines minimalen Ubuntu Linux 8.10 Intrepid Ibex Systems (mit `debootstrap`)
- Einrichten des Ubuntu Systems (hostname, fstab)
- Updaten des Ubuntu Systems
- Installieren und einrichten eines Kernels für PXE
- Installation der OpenClone Abhängigkeiten:
  - Python (python)

[35] Frei verfügbar unter `http://subversion.tigris.org/`

  - Zolera SOAP Infrastructure (python-zsi)
  - NET-3 Netzwerk Toolkit (net-tools)
  - hdparm
  - dmidecode
  - lzop
  - Partimage (partimage)
  - UDPcast (udpcast)
- OpenClone Engine installieren, einrichten
- Temporäre Dateien aufräumen

Für die Clients sollte anfangs auch Ubuntu Linux 8.04 Hardy Heron eingesetzt werden, jedoch wurde bei Tests im Schulzentrum Ybbs festgestellt, dass die On-Board Netzwerkkarte einer Fujitsu Siemens-Rechner-Serie nicht unterstützt wird. Ubuntu 8.04 setzt den Linux Kernel 2.6.24 ein. Eine genauere Analyse hat gezeigt, dass die Netzwerkschnittstelle des Intel ICH8 Chipsatzes vom Kernel 2.6.24 mit dem e1000 Treiber unterstützt werden sollte, jedoch auf diesem Mainboard Probleme machte. Weiters wurde herausgefunden, dass ab Kernel 2.6.26 die Unterstützung für die Netzwerkschnittstelle des Intel ICH8 Chipsatzes vom e1000e Treiber übernommen wurde. [Kok09] Mit einer Aktualisierung auf die Ubuntu Version 8.10 Intrepid Ibex, welche den Linux Kernel 2.6.27 einsetzt, wurde das Problem somit behoben.

NFS Server konfigurieren

Konfiguration des NFS Servers nach den zuvor abgefragten Einstellungen des Benutzers.

Pfad für Images einrichten

Der Image Speicherort wird, falls notwendig, angelegt und die entsprechenden Rechte gesetzt.

OpenClone Webservice installieren und einrichten

Das OpenClone Webservice wird installiert und die Konfigurationsdatei mit den Benutzereinstellungen gefüllt. Außerdem wird ein Datenbankbenutzer für den Zugriff vom Webservice auf die OpenClone Datenbank angelegt.

OpenClone Datenbank installieren

Anlegen der OpenClone Datenbank am MySQL Server.

OpenClone Webinterface inkl. Wrapper installieren und einrichten

Das OpenClone Webinterface wird installiert, ein Datenbankbenutzer am MySQL Server für den Zugriff des Webservices auf die OpenClone Datenbank angelegt und die Konfigurationsdatei erstellt. Die Konfiguration von `sudo` für den Webinterface Wrapper muss zur Zeit noch händisch mit dem Programm `visudo` erfolgen. (siehe Kapitel 4.5.4)

Benutzer im OpenClone Webinterface anlegen

Es wird in der OpenClone Datenbank ein Benutzer mit dem Usernamen 'admin' und dem Passwort 'openclone' angelegt.

4.6.2 PXE Umgebung

Preboot eXecution Environment ist ein Verfahren zum Booten von Rechnern über ein Netzwerk. Dazu werden verschiedene Standards eingesetzt und auch teilweise erweitert. Für die Funktion der PXE Umgebung reicht für aktuelle Rechner ein standardkonformer DHCP Server und TFTP Server, obwohl die PXE Spezifikation eine Erweiterung der Protokolle vorsieht. Bisher konnten alle Rechner im Schulnetzwerk über die standardmäßige DHCP/TFTP Lösung ohne PXE Erweiterung booten, daher gab es bisher keinen Grund auf andere Lösungen umzusteigen.

PXE Bootvorgang

Der Bootvorgang funktioniert folgendermaßen:

- Client: Anfragen nach verfügbaren Konfigurationen mit einer `DHCPDISCOVER` Nachricht
- Server: Antworten mit einer `DHCPOFFER` Nachricht, welche die IP-Adresskonfiguration, sowie einen Boot Dateinamen und den TFTP Server enthält
- Client: Anfordern der IP-Adresse mit einer `DHCPREQUEST` Nachricht
- Server: Bestätigung mit einer `DHCPACK` Nachricht, welche wiederum alle Konfigurationsoptionen enthält
- Client: Anfordern der vom DHCP Server angegebenen Boot Datei vom angegebenen TFTP Server
- Server: Senden der Boot Datei
- Client: Ausführen der Boot Datei, welche das NBP[36] enthält

Bootloader PXELINUX

Beim OpenClone System wird als NBP der Bootloader PXELINUX verwendet, welcher folgendermaßen vorgeht:

[36] Network Bootstap Program

- Laden der Konfigurationsdatei *pxelinux.cfg* vom TFTP Server
- Laden von zusätzlichen Dateien, wenn notwendig, wie z.B. das Menü
- Je nach Optionen wird sofort ein Betriebssystem gestartet, eine Kommandozeile angezeigt oder ein Menü angezeigt

PXELINUX lädt dabei nicht nur eine einzige Konfiguration vom TFTP Server sondern versucht zuerst speziellere Konfigurationen zu laden, welche nur für den einen Rechner oder eine Gruppe von Rechnern gedacht ist. Zuerst wird versucht eine Konfiguration mit der MAC-Adresse der Netzwerkschnittstelle zu finden, anschließend mit der IP-Adresse und dann mit immer größer definierten Subnetzen. Falls keine spezielle Konfiguration vorhanden ist wird die Standardkonfiguration geladen.

Linux Netzwerkboot

Das Starten von Linux über Netzwerk erfolgt folgendermaßen:

- Der Bootloader PXELINUX lädt den Kernel und das zugehörige *initrd* vom TFTP Server
- Starten des Linux Kernels
- Initialisierung des Kernels und laden der benötigten Treiber
- Abgabe der Kontrolle an das im *initrd* enthaltene `init` Script
- Konfiguration des Netzwerkinterfaces mit DHCP
- Mounten des Root-Dateisystems über NFS
- Abgabe der Kontrolle an das `init` Programm im Root-Dateisystem
- Starten der Systemdienste
- Starten der OpenClone Engine

Erweiterte DHCP Server Konfiguration

Um eine flexiblere DHCP Server Konfiguration zu erreichen, kann eine Unterscheidung zwischen den verschiedenen DHCP Clients anhand der Client Kennung (Vendor class identifier) erfolgen. Beim PXE Bootvorgang verwendet der DHCP Client immer eine Client Kennung die mit `PXEClient` beginnt. Im *initrd* wird für die Konfiguration der Netzwerkschnittstelle das Programm `ipconfig` verwendet, welches für DHCP die Client Kennung `Linux ipconfig` verwendet. Wenn der DHCP Server so konfiguriert wird, dass er nur Anfragen von diesen beiden Client Kennungen beantwortet, werden anderen DHCP Clients im Netz, wie etwa der Windows DHCP Client nicht beeinflusst. Weiters wird das Programm `ipconfig` unter Linux üblicherweise nur dann eingesetzt, wenn das Root-Dateisystem auf einem NFS Server liegt, also werden auch die normalerweise unter Linux verwendeten DHCP Clients, wie der ISC DHCP Client, nicht beeinflusst. So könnte ein anderer DHCP Server die Beantwortung von Anfragen anderer DHCP Clients übernehmen.

Kapitel 5

Ergebnisse

5.1 Analyse des Quellcodes

Zur Analyse des im Rahmen der Diplomarbeit entwickelten Source Codes wurde das Programm SLOC-Count[1] von David A. Wheeler eingesetzt, es ermittelt die Anzahl der Zeilen und stellt einige Berechnungen zum ungefähren Zeitaufwand, der Projektdauer und den möglichen Kosten an. Dabei ignoriert SLOCCount von selbst automatisch generierte Dateien, wie etwa die Webservice Client und Server Module. Leerzeilen und Kommentare werden ebenfalls ignoriert, somit beinhaltet die Statistik wirklich nur effektiv geschriebene Quellcode Zeilen (SLOC[2]). Die in Tabelle 5.1, Tabelle 5.2 und Tabelle 5.3 präsentierten Ergebnisse sind mit den Standardoptionen von SLOCCount erstellt worden.

SLOC	Verzeichnis	SLOC pro Sprache
2578	openclone-webinterface	php=2566, sh=12
1614	openclone-engine	python=1610, sh=4
825	openclone-webservice	python=751, sh=74
647	openclone-installer	sh=647
0	openclone-database	(none)

Tabelle 5.1: SLOC pro Verzeichnis

Sprache	SLOC	Prozent
php	2566	45,30%
python	2361	41,68%
sh	737	13,01%

Tabelle 5.2: SLOC Summe pro Sprache

[1] Frei verfügbar unter `http://www.dwheeler.com/sloccount/`
[2] Source Lines Of Code

Summe effektiver Quellcodezeilen (SLOC)	5.664
Entwicklungsaufwand geschätzt, Personenjahre (Personenmonate) (COCOMO-Basismodell, $Personenmonate = 2,4 * (KSLOC^{1,05})$)	1,24 (14,82)
Projektdauer geschätzt, Jahre (Monate) (COCOMO-Basismodell, $Monate = 2,5 * (Personenmonate^{0,38})$)	0,58 (6,96)
Geschätzte durchschnittliche Anzahl an Entwickler ($Aufwand/Dauer$)	2,13
Geschätzte Gesamt-Entwicklungskosten (Durchschnittliches Gehalt = US$ 56.286/Jahr, Overhead = 2,40)	US$ 166.886

Tabelle 5.3: SLOCCount Statistik

5.2 Einsatz und Installation im Schulzentrum Ybbs

Für den Einsatz im Schulzentrum Ybbs wurde eine virtuelle Maschine mit Ubuntu Linux 8.04 Hardy Heron und der OpenClone Software eingerichtet. Die virtuelle Maschine läuft unter VMware Server 2[3].

Im Schulnetzwerk sind die Computer nach den Räumen in Subnetze eingeteilt. Um von jedem gewünschten Raum mit der virtuellen Maschine kommunizieren zu können, muss für diese im jeweiligen Subnetz eine IP-Adresse vergeben werden. Weiteres werden die IP-Adressen der Computer von einem zentralen DHCP-Server vergeben. Um das Booten über Netzwerk mit PXE von der virtuellen Maschine aus zu ermöglichen, musste am DHCP-Server der Schule die `next-server` Option für jedes Subnetz auf die jeweilige IP-Adresse der virtuellen Maschine eingestellt werden. Die Konfiguration von zusätzlichen Subnetzen für OpenClone wird im Anhang B.3.1 beschrieben.

Zur Absicherung der Kommunikation zwischen Verwaltungsclient und OpenClone Server wurde in der virtuellen Maschine der HTTP Server Apache mit einer SSL Konfiguration erweitert. Das Webinterface ist jedoch weiterhin über den unverschlüsselte HTTP Protokoll erreichbar. Die Konfiguration von HTTPS wird im Anhang B.3.2 beschrieben.

5.3 Veröffentlichung als OpenSource Projekt

Um den Anwendungsbereich des Cloningsystems zu erweitern, wurde das Cloningsystem unter eine OpenSource Lizenz gestellt und frei zugänglich gemacht. Im Kapitel 5.3.1 werden die notwendigen Schritte zur Lizenzierung und weiters im Kapitel 5.3.2 der Registrierungsprozess bei einem OpenSource Projekthosting Anbieter beschrieben. Im Kapitel 5.3.3 wird auf die zukünftig geplanten Erweiterungen des Projektes eingegangen.

5.3.1 Lizenzierung

Eine Anforderung der Diplomarbeit war die Lizenzierung der entwickelten Software unter einer OpenSource Lizenz. Die Entscheidung fiel auf die weit verbreitete GNU General Public License in der Version 3[4], welche jedem die Nutzung, auch für kommerzielle Zwecke, erlaubt. Auch Veränderungen sind je-

[3] Als Freeware verfügbar unter `http://www.vmware.com/de/products/server/`
[4] Lizenztext verfügbar unter http://www.gnu.org/licenses/gpl.html

dem erlaubt, jedoch wird bei jeglicher Verbreitung einer veränderten sowie der Original-Version die Offenlegung des Quellcodes gefordert.

Zur Lizenzierung sind folgende Schritte notwendig:

- Einfügen einer Lizenzdatei ins Projektverzeichnis mit dem gesamten Lizenztext
- Einfügen eines Lizenz Headers in jede Quellecodedatei und wenn möglich auch in Mediendateien
- Angabe des Copyrights in jeder Quellcode- und Mediendatei

5.3.2 Hosting

Der Hostinganbieter stellt alle für das Projekt benötigten Dienste zur Verfügung, dazu zählt für unser Projekt hauptsächlich die Versionverwaltung Subversion zur Quellcodeverwaltung und eine Webseite zur Präsentation des Projektes.

Obwohl es zahlreiche kostenfreie Hostinganbieter gibt, fiel die Entscheidung auf den Dienst Savannah[5] der Free Software Foundation, da dieser komplett unabhängig von einzelnen Unternehmen ist. Außerdem basiert der Dienst Savannah selbst rein auf OpenSource Software.

Das OpenClone Projekt wurde bei Savannah eingereicht und unterläuft zur Zeit eine Aufnahmeüberprüfung. Wenn alle Anforderungen erfüllt sind und das Projekt freigeschalten wird, sollte die Website bald unter der URL `http://www.nongnu.org/openclone/` erreichbar sein.

Der Registrierungsprozess läuft folgendermaßen ab:

- Registrierung als Benutzer
- Einreichen des Projektes mit Kurzbeschreibung und Quellcode
- Entscheidung der Freischaltung abwarten und gegebenfalls noch kleine Änderungen vornehmen, damit das Projekt allen Anforderungen genügt
- Einrichten der vom Projekt benötigten Dienste

5.3.3 Roadmap

Im folgenden Abschnitt wird auf die für bestimmte Versionen gesteckten Meilensteine eingegangen:

- Version 0.1.5
  - Löschen von Hosts
  - Unterstützung von Rechnern mit Phonix Card von EduSystem (ehem. RebornCard von Signal Computer GmbH)[6]

[5] Erreichbar unter `http://savannah.nongnu.org/`

[6] Die Phonix bzw. Reborn Card ist eine in Hardware implementierte Lösung, um den Festplatteninhalt eines Computers nach einem Neustart auf einen definierten Zustand zurückzusetzen.

- Version 0.2.0
  - Berechnung des Gesamtfortschritts und Aktualisierung der Statusinformationen mit AJAX
  - Unterstützung der Imagingsoftware partclone (bietet ext4 Unterstützung)
  - Unterstützung von unbekannten Dateisystemen durch 1:1 Kopien mit dd
- Version 0.3.0
  - Unterstützung von Post-Cloning Operationen (Hostname ändern, Windows Domain beitreten, ...)
- Version 0.5.0
  - Erweiterter Installer
  - Wake-On-LAN Unterstützung
- Version 1.0.0
  - Unterstützung von Clonen mehrerer Festplatten und einzelnen Partitionen
- Zukunft
  - Clonezilla/FOG Images importieren
  - Unterstützung von mehreren Fileservern
  - Webserviceschnittstelle zur Steuerung des Cloningsystems
  - BIOS Einstellungen clonen

Zusammenfassung

Das Projekt befasste sich mit einem Vergleich existierender Cloninglösungen und der Entwicklung eines eigenen Systems basierend auf den gewonnen Informationen aus der Analyse. Alle Pflichtanforderungen und einige Wunschanforderungen wurden erfolgreich implementiert. Das System konnte im Schulzentrum Ybbs a.d. Donau getestet werden und wird dort auch für den praktischen Einsatz installiert.

Bei der Entwicklung wurde besonders auf die Bedürfnisse des Schulzentrums eingegangen, um die Software für den dortigen Gebrauch zu optimieren. Weiters wurde auf einfache Handhabungsmöglichkeit und Modularität geachtet. Somit ist eine Substitution einzelner Teilkomponenten oder das Einpflegen weiterer Softwareteile ohne größere Umstände realisierbar. Da die Implementierung teilweise mit neuen Programmiersprachen und Techniken erfolgte, konnten in diesen Bereichen wertvolle neue Erfahrungen gesammelt werden.

Durch die Verwendung einer freien Softwarelizenz und der Veröffentlichung als Open Source Projekt wird die entwickelte Lösung auch anderen Schulen und Institutionen zur Verfügung gestellt. Weiters ist es Interessenten dadurch möglich, die Software nach eigenem Ermessen anzupassen und zu verbessern, um sie optimal auf individuelle Einsatzbedingungen abstimmen zu können. Eine zukünftige Weiterentwicklung wird auch von Seiten des Projektteams selbst erfolgen. Für die Implementierung von Funktionen in naher Zukunft wurde eine Roadmap erstellt. Diese ist im Kapitel 5.3.3 einzusehen.

Literaturverzeichnis

[AA09] ATA-ATAPI.COM. How It Works: Partition Tables. http://ata-atapi.com/hiwtab.html, April 2009.

[BM09] Joshua Boverhof and Charles Moad. ZSI: The Zolera Soap Infrastructure - User's Guide. http://sourceforge.net/project/downloading.php?group_id=26590&filename=guide-zsi-2.0.pdf&a=54532226, April 2009.

[Bov09] Joshua Boverhof. ZSI - RPC Style binding, extended generation. http://osdir.com/ml/python.pywebsvcs.general/2005-10/msg00005.html, April 2009.

[Bra09] Pádraig Brady. Details of GRUB on the PC. http://www.pixelbeat.org/docs/disk/, April 2009.

[Com09a] Hewlett-Packard Company. Management and Configuration Guide for the HP ProCurve Switches 1600m, 2424m, 4000m, and 8000m. ftp://ftp.hp.com/pub/networking/software/59692320.pdf, April 2009.

[Com09b] Hewlett-Packard Company. ProCurve Series 2500 Switches Management and Configuration Guide. ftp://ftp.hp.com/pub/networking/software/2500-MgmtConfig-Oct2005-59692354.pdf, April 2009.

[Doc09] Doctrine. Doctrine 1.1 Documentation. http://www.doctrine-project.org/documentation, April 2009.

[Fre09] Free Software Foundation, Inc. Gzip User's Manual. http://www.gnu.org/software/gzip/manual/index.html, April 2009.

[Gmb09] Kortenbrede Datentechnik GmbH. Symantec Ghost Solution Suite 1.1 EDU Preis. http://www.educheck.de/shop/shop39974-h1291.htm, April 2009.

[GS09] S. Gerhold and M. Sonnenfroh. Betriebssysteme SS07 - Kapitel 2: Booten. https://www-vs.informatik.uni-ulm.de/wiki/pub/Main/BetriebsSystemeSS07/BS_Kapitel2_print.pdf, April 2009.

[Inc09a] Acronis Inc. Acronis Snap Deploy 3 für Workstation. http://www.acronis.de/enterprise/products/snapdeploy/, April 2009.

[Inc09b] Acronis Inc. Acronis Snap Deploy 3 Handbuch. http://download.acronis.com/pdf/UG_ASD3_DE.pdf, April 2009.

[Kok09] Auke Kok. [ANNOUNCE] e1000 to e1000e migration of PCI Express devices. `http://article.gmane.org/gmane.linux.drivers.e1000.devel/3601`, April 2009.

[LNP09] Linux-NTFS-Project. ntfsclone. `http://www.linux-ntfs.org/doku.php?id=ntfsclone`, April 2009.

[Obe09] Markus Franz Xaver Johannes Oberhumer. LZO – a real-time data compression library. `http://www.oberhumer.com/opensource/lzo/lzodoc.php`, April 2009.

[Par09a] Partclone. Partclone. `http://partclone.nchc.org.tw/`, April 2009.

[Par09b] Partimage. Supported-Filesystems. `http://partimage.org/Supported-Filesystems`, April 2009.

[SB09] Rich Salz and Christopher Blunck. ZSI: The Zolera Soap Infrastructure - Developer's Guide. `http://sourceforge.net/project/downloading.php?group_id=26590&filename=zsi-2.0.pdf&a=6917220`, April 2009.

[Sed09] Daniel B. Sedory. MBR/EBR Partition Tables. `http://www.geocities.com/thestarman3/asm/mbr/PartTables2.htm`, April 2009.

[Sew09] Julian Seward. bzip2 Documentation. `http://www.bzip.org/docs.html`, April 2009.

[SQL09] SQLAlchemy. SQLAlchemy 0.4 Documentation. `http://www.sqlalchemy.org/docs/04/`, April 2009.

[UDP09] UDPcast. Udpcast commandline options. `http://udpcast.linux.lu/cmd.html`, April 2009.

[Wik09a] Wikipedia. Lempel-Ziv-Markow-Algorithmus. `http://de.wikipedia.org/wiki/Lempel-Ziv-Markow-Algorithmus`, April 2009.

[Wik09b] Wikipedia. Web Services Description Language. `http://de.wikipedia.org/wiki/Web_Services_Description_Language`, April 2009.

[Wil09] Brendon J. Wilson. Ruby + SOAP4R + WSDL Hell. `http://www.brendonwilson.com/blog/2006/04/02/ruby-soap4r-wsdl-hell/`, April 2009.

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Anhang A

Projekt

Dieser Abschnitt enthält die Anforderungsspezifikation, den Projektplan, Verantwortlichkeiten und Sitzungsprotokolle aus dem Bereich Projektmanagement.

A.1 Anforderungsspezifikation

In diesem Abschnitt sind Pflicht- sowie Wunschanforderungen und Ausgrenzungskriterien aufgelistet. Weiters werden Kriterien angeführt, die von den Diplomanden selbst definiert wurden.

Pflichtanforderungen

Die Pflichtanforderungen umfassen folgende Punkte:

- Vergleich von bestehenden Cloning Systemen mit Berücksichtigung der Anforderungen des Schulzentrums Ybbs a.d. Donau
- Entwickeln eines Cloning Systems, welches den Anforderungen des Schulzentrums genügt
  - Es muss möglich sein, von Ziel-Rechnern mit den Betriebssystemen Windows (XP, Vista), Linux (SuSE 11) eine komplette Sicherung einer Festplatte zu erstellen und diese auf einem anderen Ziel-Rechner wiederherzustellen
  - Es muss auch eine Kombination von einem Windows und Linux Betriebssystem (Dual Boot Systeme) gesichert und wiederhergestellt werden können
  - Die Sicherungen sollen auf einem zentralen Server gespeichert werden
  - Es soll eine parallele Wiederherstellung auf mehrere Ziel-Rechner mittels Multicast möglich sein
  - Die benötigte Zeit für eine Sicherung und Wiederherstellung soll im Vergleich zur aktuellen Lösung mit Norton Ghost nicht erheblich länger sein. (maximal doppelt so lange)
  - Auf den Ziel-Rechnern soll keine zusätzliche Software installiert werden müssen
  - Die Ziel-Rechner sollen alle benötigten Daten über das Netzwerk beziehen

  - Die Kommunikation soll über die bestehende Netzwerkhardware erfolgen
  - Die Integration in das bestehende Netzwerk soll möglich sein (zentraler DHCP-Server, Subnetz für jeden Computerraum)
  - Die Steuerung (Sicherung, Wiederherstellung starten) des Cloning Systems soll über eine zentrale Software möglich sein
  - Der Status des Sicherungs-/Wiederherstellungsprozesses jedes Ziel-Rechners soll mittels der zentralen Software abgerufen werden können
  - Die zentrale Software soll einfach zu bedienen sein. (z.B. durch Wizards)
  - Der Zugriffe auf die Steuerungssoftware sollen durch Passwort-Authentifizierung und Verschlüsselung gesichert sein
  - Die Ziel-Rechner sollen am zentralen Server mit dem Hostnamen, IP-Adresse, MAC-Adresse, Seriennummer gespeichert werden
  - Die Ziel-Rechner sollen einer Gruppe zugewiesen werden können
  - Das Hinzufügen neuer Ziel-Rechner soll bis auf Hostname, IP-Adresse, Seriennummer, Gruppe automatisch erfolgen
  - Der Hostname, IP-Adresse, Seriennummer sollen über die zentrale Software eingetragen werden können
  - Die Wiederherstellung einer Sicherung soll auf eine ganze Gruppe möglich sein
  - Auf dem zentralen Server gespeicherte Sicherungen sollen über die zentrale Software gelöscht werden können
  - Die Ziel-Rechner sollen nach einem erfolgreichen Cloningvorgang heruntergefahren werden können
  - Es sollen möglichst keine Lizenzkosten anfallen.

- Installation des Cloning Systems im Schulzentrum

Wunschanforderungen

- Die benötigte Dauer für eine Sicherung und Wiederherstellung soll kürzer als die der aktuellen Lösung mit Norton Ghost sein
- Zusammenfassende Darstellung der Statusinformationen eines ganzen Sicherungs- und Wiederherstellungsprozesses in der zentralen Software
- Hostname, IP-Adresse von mehreren Rechnern gleichzeitig über eine Erzeugungsregel festlegen
- Speicherung von zusätzlichen Hardware Informationen am zentralen Server
- Automatisierung der Post-Cloning Operationen (Setzen des Hostnamens, Beitreten der Windows Domain)
- Erkennung der installierten Betriebssysteme

Ausgrenzungskriterien

- Sicherung und Wiederherstellung von einzelnen Partitionen eines Ziel-Rechners
- Sicherung und Wiederherstellung von mehreren Festplatten eines Ziel-Rechners in einem einzigen Durchlauf
- Speicherung der Sicherungen an einem anderen Ort, als auf der internen Festplatte des zentralen Servers
- Automatisches Einschalten (Power on) der Ziel-Rechner
- Direkte Übertragung der Daten von einem Quell-Rechner zu Ziel-Rechnern ohne Zwischenspeicherung am zentralen Server
- Absicherung (Authentifizierung, Verschlüsselung) der Kommunikation zwischen Server und Ziel-Rechner
- Wartung des laufenden Systems

Eigene Kriterien der Diplomanden

- Die Verwaltungssoftware soll plattformübergreifend verwendet werden können
- Es soll die Sicherung und Wiederherstellung von Ziel-Rechnern mit der Linux Distribution Ubuntu unterstützt werden
- Das Cloning System soll nur von OpenSource Software abhängig sein
- Das Cloning System soll unter einer Open Source Lizenz, wie der GNU General Public License veröffentlicht werden

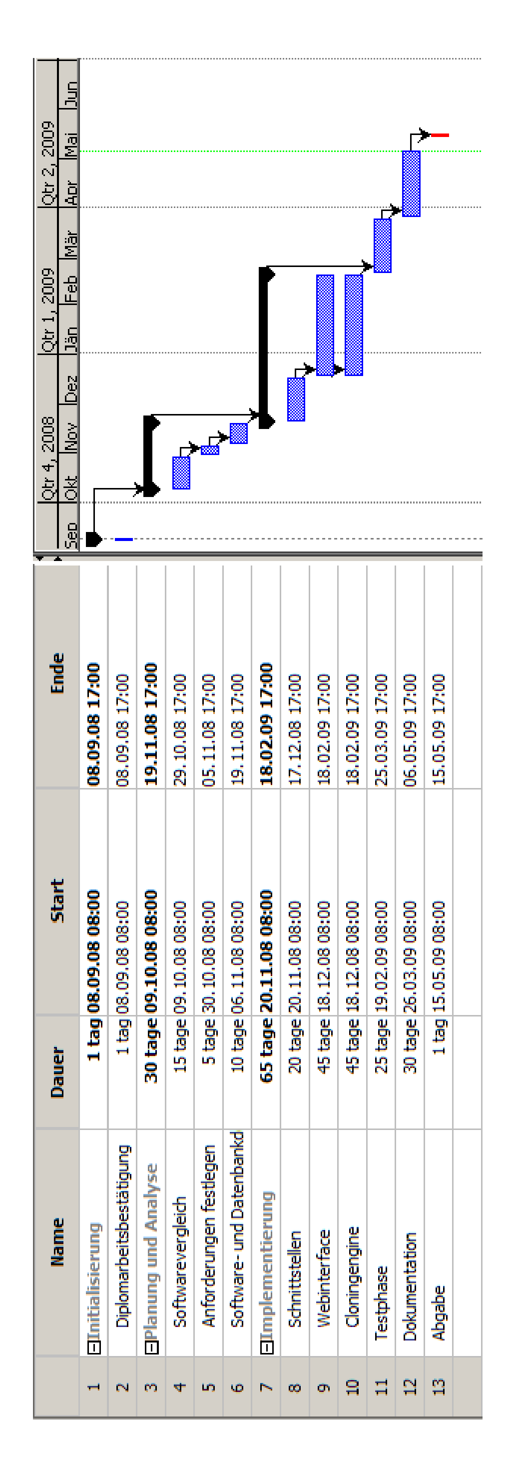
A.2 Projektplan

In der Abbildung A.1 ist der Projektablauf grafisch in Form eines Gantt Diagrammes veranschaulticht. Weiters sind die einzelnen Phasen mit der Gesamtdauer, dem Start- und dem Enddatum ersichtlich. Der Zeitplan wurde mit dem Programm OpenProj[1] erstellt.

A.3 Verantwortlichkeiten

Im Folgenden wird die Aufteilung der Verantwortlichkeiten erläutert. Die Analyse, welche den Vergleich von Cloningsystemen beinhaltete, wurde von beiden Diplomanden gemeinsam durchgeführt. Nach dem gemeinsamen Entwurf der Datenbank wurde die Arbeit so aufgeteilt, dass ein sinnvolles, von einander weitestgehend unabhängiges Arbeiten für jedes Projektmitglied möglich war. David Gnedt war für die

[1] Frei verfügbar unter `http://openproj.org/`



	Name	Dauer	Start	Ende
1	**Initialisierung**	**1 tag**	**08.09.08 08:00**	**08.09.08 17:00**
2	Diplomarbeitsbestätigung	1 tag	08.09.08 08:00	08.09.08 17:00
3	**Planung und Analyse**	**30 tage**	**09.10.08 08:00**	**19.11.08 17:00**
4	Softwarevergleich	15 tage	09.10.08 08:00	29.10.08 17:00
5	Anforderungen festlegen	5 tage	30.10.08 08:00	05.11.08 17:00
6	Software- und Datenbankd	10 tage	06.11.08 08:00	19.11.08 17:00
7	**Implementierung**	**65 tage**	**20.11.08 08:00**	**18.02.09 17:00**
8	Schnittstellen	20 tage	20.11.08 08:00	17.12.08 17:00
9	Webinterface	45 tage	18.12.08 08:00	18.02.09 17:00
10	Cloningengine	45 tage	18.12.08 08:00	18.02.09 17:00
11	Testphase	25 tage	19.02.09 08:00	25.03.09 17:00
12	Dokumentation	30 tage	26.03.09 08:00	06.05.09 17:00
13	Abgabe	1 tag	15.05.09 08:00	15.05.09 17:00

Abbildung A.1: Zeitplan des Projektverlaufes

Entwicklung des Webservices und der Engine zuständig. Manuel Steiner implementierte parallel dazu das Webinterface für die Steuerung des Cloningsystems. Das Testen und die Installation der Lösung im Schulzentrum erfolgte wieder in gemeinsamer Arbeit.

A.4 Sitzungsprotokolle

Dieser Abschnitt beinhaltet alle Protokolle der abgehaltenen Sitzungen, in denen der besprochene Inhalt und die zu erledigenden Aufgaben festgehalten wurden.

Sitzungsprotokoll vom 3.12.2008

Teilnehmer

- DI Andreas Brachinger
- David Gnedt
- Manuel Steiner

Ort

IT-HTL Ybbs a.d. Donau

Besprochenes

Es wurde bisherige Überlegungen zum Softwaredesign vorgestellt.

- Die Software, welche die Schnittstelle zum Benutzer darstellt, soll als Webinterface implementiert werden. Dazu würden sich die Sprachen PHP oder JavaScript eignen.
- Die Engine, die auf den Clients läuft, soll über ein Webservice angebunden werden. Hier wären Ruby, Python oder PHP als Sprachen denkbar.
- Entweder wird die Software zweiteilig oder dreiteilig implementiert. Bei einer Splittung in zwei Teile greift die Engine auf den Web Service Server zu, welcher dann auf der Datenbank operiert. Das Webinterface greift direkt auf die Datenbank zu. Bei einer Dreiteilung greift das Webinterface ebenfalls über ein Webservice auf die Datenbank zu. Dazu wären zwei Web Service Server nötig.
- Die Programmiersprachenwahl können die Diplomanden selbst treffen. Java und C# erscheinen wegen einer eigenen virtuellen Maschine eher ungeeignet. Perl hat eine Syntax, mit der man sich erst anfreunden müsste. Ruby sollte einfach zu lernen sein, jedoch ist hier keine Hilfestellung von Seiten der Diplomarbeitsbetreuer möglich. Python wäre ebenfalls noch eine Alternative.
- Es soll ein externer DHCP Server unterstützt werden, weil der DHCP Server auf der Firewall betrieben wird, das Cloningsystem jedoch aufgrund seines Umfanges auf einen anderen Server ausgelagert werden soll. Die DHCP Serverkonfiguration wäre über SSH möglich. Ein Export der Konfigurationsdatei aus den Informationen im Webinterface würde genügen. Es wäre auch akzeptabel, die Konfiguration nicht automatisch zu erzeugen, weil eine Änderung dieser nicht sehr häufig erfolgt.
- Einige Befehle müssen im System als Benutzer root ausgeführt werden. Dazu wäre es möglich, einen Public Key Exchange für SSH durchzuführen und danach auf localhost zu verbinden, oder man verwendet das sudo Kommando und schränkt die Berechtigungen so ein, dass bestimmte Benutzer mit Rootrechten bestimmte Skripte ausführen dürfen. Dies wäre die bessere Lösung

Zu Erledigendes

Es soll mit Herrn Professor Ungerböck über Webservices gesprochen werden, weil er in diesem Gebiet viel Erfahrung hat.

Sitzungsprotokoll vom 5.12.2008

Teilnehmer

- DI (FH) Georg Ungerböck
- David Gnedt
- Manuel Steiner

Ort

IT-HTL Ybbs a.d. Donau

Besprochenes

Es wurden verschiedene Programmiersprachen zu Webservices verglichen. Für den Einsatz als SOAP Server ist PHP eher ungeeignet, weil es zu wenige Funktionen bietet. Ein Webservice zwischen Datenbank und Webinterface ist eventuell zu aufwändig und zu zeitintensiv. SOAP sollte mit Python gut funktionieren. Mit Ruby hat Herr Professor Ungerböck selbst noch keine Erfahrung. Der Einsatz von WSDL ist nicht unbedingt empfehlenswert. Man sollte die Funktionen am Client und Server festlegen. Bei einer Änderung des Protokolls wird meistens ein komplett neues Web Service generiert und das alte aus Kompatibilitätsgründen parallel weiter betrieben.

Sitzungsprotokoll vom 17.12.2008

Teilnehmer

- DI Andreas Brachinger
- DI Christian Hammer
- David Gnedt
- Manuel Steiner

Ort

IT-HTL Ybbs a.d. Donau

Besprochenes

Es wurde je ein Webservice Beispiel in Python und Ruby vorgezeigt. Danach wurde die Funktionsweise des Cloningsystems besprochen.

- Systemstart
  - In der Datenbank eingetragene Rechner werden nur über PXE gestartet, wenn Aufgaben für ihn verfügbar sind.
  - Neue Rechner werden nur über PXE gestartet, wenn die automatische Registrierung im Webinterface aktiviert ist.
  - Wenn ein Rechner über PXE gestartet wird, und das Cloning System nicht gestartet ist, soll das lokale Betriebssystem am Client geladen werden. Alternativ könnte man vom DHCP Server keine Antwort schicken oder dem Client kein Bootimage schicken. Dies würde jedoch zu einer größeren Verzögerung des Bootvorganges führen.
- Zwischen Cloningserver und -client wird ein Webservice eingesetzt, dessen Funktion in einer WSDL Datei beschrieben wird. Mittels Hilfswerkzeugen können aus dieser Datei sowohl bei Python als auch bei Ruby Funktionsgrundgerüste erstellt werden, mit denen sehr einfach ein Client oder Server programmiert werden kann.
- Der DHCP Server kann auf verschiedenen Systemen laufen. Entweder er wird nur am Cloning Server verwendet oder es können zwei DHCP Server eingesetzt werden. Einer ist für normale DHCP Anfragen zuständig und der zweite DHCP Server beantwortet nur DHCP Anfragen von PXE Clients. Dazu müssten die beiden DHCP Server aber so konfiguriert werden, dass sie nur die jeweiligen Anfragen beantworten.
- Wenn alle Clients als ersten Booteintrag PXE eingestellt hätten, könnte Wake on LAN verwendet werden, um nicht jeden Client händisch einschalten zu müssen. In der Schule sind in den meisten Computerräumen die Rechner jedoch durch einen Schlüsselschalter vom Stromnetz getrennt. Somit muss erst der Strom aufgedreht werden, bevor die Computer gebootet werden können. Damit ist Wake on LAN hinfällig.

Sitzungsprotokoll vom 7.1.2009

Teilnehmer

- DI Andreas Brachinger
- DI Christian Hammer
- David Gnedt
- Manuel Steiner

Ort

IT-HTL Ybbs a.d. Donau

Besprochenes

Es wurde der Datenbankentwurf besprochen. Es wäre noch ein Feld für die Seriennummer der Computer optimal. Die Attribute für ein Geräte eines Computers werden automatisch am Client ausgelesen und bei der Registrierung oder beim Login am Server in die Datenbank eingetragen.
Weiters wurde der aktuelle Status erläutert. Das Clonen einer kompletten Festplatte wird funktionieren, jedoch wird wahrscheinlich zu wenig Zeit sein, um Postcloningoperationen, wie zum Beispiel das Ändern des Hostnamens, zu implementieren. Dazu könnte eventuell fertige Software wie ghostwalk oder newsid verwendet werden.

Sitzungsprotokoll vom 28.1.2009

Teilnehmer

- DI Andreas Brachinger
- DI Christian Hammer
- David Gnedt
- Manuel Steiner

Ort

IT-HTL Ybbs a.d. Donau

Besprochenes

Es wurde der Imagingvorgang einer einzelnen Partition vorgeführt. Weiters waren kleine Ausbesserungen an der Datenbank nötig.

Zu Erledigendes

Bei der nächsten Besprechung, am 11.2.2009 soll ein gesamter Imaging- und Restorevorgang eines in einer virtuellen Maschine installierten Windows XP Systems funktionieren.

Sitzungsprotokoll vom 11.2.2009

Teilnehmer

- DI Andreas Brachinger
- DI Christian Hammer
- David Gnedt
- Manuel Steiner

Ort

IT-HTL Ybbs a.d. Donau

Besprochenes

Es wurde der Imaging- und Restoreprozess einer gesamten Festplatte vorgeführt. Dabei wurde ein Windows XP System in einer virtuellen Maschine verwendet. Der Festplatteninhalt wurde gesichert. Danach wurde eine leere virtuelle Festplatte vom Cloningsystem formatiert, der Bootloader und der Festplatteninhalt zurückgesichert.

Zu Erledigendes

Der Status eines Imaging- oder Restoreprozesses soll im Webinterface angezeigt werden. Weiters soll es möglich sein, bereits eingetragene Aufgaben für einen Client wieder zu löschen. Der Name eines Images soll ebenfalls im Webinterface ersichtlich sein. Die nächste Besprechung findet am 25.2.2009 statt.

Sitzungsprotokoll vom 25.2.2009

Teilnehmer

- DI Andreas Brachinger
- DI Christian Hammer
- David Gnedt
- Manuel Steiner

Ort

IT-HTL Ybbs a.d. Donau

Besprochenes

Der Rücksicherungsvorgang über Multicast sowie die Statusanzeige im Webinterface wurde vorgeführt. Eine Testinstallation des Systems soll in einer virtuellen Maschine erfolgen.

Zu Erledigendes

Im Webinterface soll eine Meldung angezeigt werden, sobald ein Vorgang beendet wurde. Weiters soll es möglich sein, die Clients nach Fertigstellung ihrer Aufgaben herunterzufahren oder neuzustarten. Eventuell soll das Multicasting mit den in der Schule eingesetzten HP Switches getestet werden. Bei bisherigen Cloningvorgängen wurde ein Multicast immer als Broadcast ausgesendet, weil Multicastgruppen am Switch händisch konfiguriert werden müssen. Die nächste Besprechung findet am 11.3.2009 statt.

Sitzungsprotokoll vom 11.3.2009

Teilnehmer

- DI Andreas Brachinger
- DI Christian Hammer
- David Gnedt
- Manuel Steiner

Ort

IT-HTL Ybbs a.d. Donau

Besprochenes

Es wurde das Webinterfacedesign sowie das Clonen einer erweiterten Partition und die Statusanzeige im Webinterface vorgezeigt. Es wurde ein praktischer Imagingtest mit einem Lenovo Netbook durchgeführt. Der Rechner konnte sich erfolgreich registrieren und der Imagingprozess startete, jedoch war am Server zu wenig Festplattenplatz vorhanden.

Zu Erledigendes

Beim Clonen von erweiterten Partitionen muss im Webinterface noch ein Fehler behoben werden. Weiters soll das System getestet werden. Dazu könnten virtuelle Machinen als Clients verwendet werden. Die Performance wäre jedoch mit richtiger Hardware um einiges besser.

Sitzungsprotokoll vom 15.4.2009

Teilnehmer

- DI Andreas Brachinger
- DI Christian Hammer
- David Gnedt
- Manuel Steiner

Ort

IT-HTL Ybbs a.d. Donau

Besprochenes

Es wurde ein Image von einem Rechner im Softwarelabor des Schulzentrums erstellt und dieses auf alle Rechner im Raum zurückgespielt. Anfangs gab es Schwierigkeiten mit der Cloningengine, weil diese die Netzwerkkarten der Computer nicht erkannte. Der Fehler konnte jedoch behoben werden.

Zu Erledigendes

Für das Webinterface wurden folgende Änderungen gewünscht.

- Bei der Auflistung der Hosts sollen alle Hosts auf einmal markiert werden können
- Es soll die Gesamtanzahl der angezeigten Hosts ersichtlich sein
- Bei der Auswahl einer anzuzeigenden Gruppe, soll ersichtlich sein, welche Gruppe gerade angezeigt wird
- Bei den durchführbaren Aktionen können Checkboxen statt Radio Buttons verwendet werden, um mehrere Aktionen gleichzeitig auszuführen
- Bei der Vergabe der Hostnamen soll die Postfixnummer dreistellig sein (z.B. 010)
- Bei den Details eines Hosts soll ebenfalls die Änderung der IP Adresse und des Hostnamen möglich sein.
- Die Größe eines Images soll ersichtlich sein
- Der Name eines Images soll veränderbar sein

Folgende eigene Ziele sind noch zu implementieren.

- Nach dem ein Wizard fertig durchgegangen wurde, soll ein Link auf die Statusseite angezeigt werden
- Auf der Statusseite soll in regelmäßgen Abständen ein automatischer Seitenrefresh erfolgen
- Verwendung von Transaktionen bei Datenbankaktionen

Sitzungsprotokoll vom 24.4.2009

Teilnehmer

- DI Andreas Brachinger
- DI Christian Hammer
- David Gnedt
- Manuel Steiner

Ort

IT-HTL Ybbs a.d. Donau

Besprochenes

Das Blockschaltbild über den generellen Aufbau des Systems und das EER Diagramm der Datenbank wurde abgegeben. Weiters wurden alle Änderungswünsche, welche in der letzten Besprechung geäußert wurden, implementiert. Diese wurden vorgezeigt. Weiters ist es möglich zwischen den Kompressionsverfahren gzip und lzo auszuwählen. Falls in einem Raum neue Rechner installiert werden, die dort zuvor eingesetzten Computer in einen anderen Raum transportiert werden und die dort eingesetzten Rechner nicht mehr verwendet werden, ist folgende Vorgehensweise zu empfehlen.

1. Die Rechner, welche nicht mehr verwendet werden, im Webinterface aus der jeweiligen Gruppe löschen oder in eine andere Gruppe verschieben. Die Hostnamen und IP Adressen müssen auf nicht benutzte Werte geändert werden.
2. Den Computern, welche von einem Raum in einen anderen übersiedeln, die in dem neuen Raum verwendeten IP Adressen und Hostnamen zuweisen und diese Computer in die entsprechende Gruppe verschieben.
3. Die neuen Rechner in die Gruppe verschieben, die dem Raum zugeordnet ist, in dem sie installiert werden und die IP Adressen und Hostnamen ändern.

Zu Erledigendes

Es wurde der Wunsch geäußert, Hosts aus dem Cloningsystem löschen zu können. Diese Funktion wird allerdings erst nach der Diplomarbeit implementiert.

Anhang B

Benutzerhandbuch

B.1 Installationsanleitung

Für die einfache Installation des OpenClone Systems auf einem Server steht ein Installationsskript zur Verfügung. Dieses Skript installiert die Software und alle benötigten Dienste auf dem Server. Damit das Skript ordnungsgemäß ausgeführt wird, muss es mit Rootrechten aufgerufen werden. Weiters wird das Betriebssystem Ubuntu 8.04 LTS benötigt. Diese Version bietet einen längeren Supportzeitraum als andere Distributionen, wie zum Beispiel ein normales Ubuntu Linux.

Nach dem erstmaligen Aufruf des Skripts mittels `sudo ./install.sh` wird man gefragt, ob man OpenClone installieren möchte. Mit der Eingabe von `y` wird diese Abfrage bestätigt und die Installation beginnt. Danach wird gefragt, über welches Netzwerkinterface der Server mit den Clients kommunizieren soll. Standardmäßig ist hier eth0 eingestellt. Sollte hier ein anderes Interface verwendet werden, ist der Name dieses Interfaces einzugeben. Danach muss man die IP Adresse für das Interface eingeben. In weiterer Folge werden noch die Adresse des Netzwerkes, die Netzmaske und die Anzahl der Bits der Netzmaske abgefragt. Danach folgt die Konfiguration für den DHCP Server. Es sind die Start- und Endadresse für den Bereich anzugeben, welcher vom DHCP Server dynamisch an die Clients vergeben werden darf.

Nach dieser Konfiguration folgt jene des TFTP Servers. Man muss das Verzeichnis angeben, in dem das PXE Bootimage für die Clients liegt. Standardmäßig wird hier /tftpboot verwendet. Anschließend wird das Wurzelverzeichnis für den NFS Serverdienst abgefragt. Hier ist /nfsroot voreingestellt. Ist diese Einstellung vorgenommen, muss noch das Verzeichnis angegeben werden, in dem die Festplattenimages der Clients gespeichert werden sollen. Für diesen Wert wird /data als Standard angenommen.

Sind all diese Einstellungen vorgenommen, wird die Installation des Systems durchgeführt. Es muss gegebenfalls die Erlaubnis gegeben werden, Softwarepakete vom Internet nachladen zu dürfen. Bei der Installation des MySQL Datenbankservers wird das Passwort des Benutzers root für den Zugriff auf den Dienst abgefragt. Dieses Passwort muss ein zweites Mal zur Bestätigung eingegeben werden. Bei erfolgreicher Installation erscheint die Zeile `*** Installation finished ***`.
Folgende notwendige Einstellungen werden noch nicht vom Installationsskript durchgeführt. Diese sind händisch vorzunehmen.

- Für die Einrichtung des Datenpfades des TFTP Servers muss die Datei /etc/inetd.conf entsprechend angepasst werden

- Die Konfiguration für den Webinterface Wrapper muss noch händisch erledigt werden. Mit dem Programm visudo muss die Datei /etc/sudoers so angepasst werden, dass der Benutzer, unter dem der Apache Prozess läuft, ohne Passwortabfrage das Webinterface Wrapper Skript mit Rootrechten ausführen darf

Das Installationsskript kann auch für ein Update der OpenClone Software genutzt werden. Dazu wird das Skript erneut mit Rootrechten aufgerufen. Die Frage, ob die Software auf den aktuellsten Stand gebracht werden soll, ist mit `y` zu beantworten. Nach dieser Abfrage wird ein OpenClone Update durchgeführt. Bei Erfolg wird die Zeile `*** Update finished ***` ausgegeben.

Weitere Informationen zum Installationsskript sind im Kapitel 4.6 zu finden.

B.2 Handhabung des Webinterface

Dieser Abschnitt beschreibt die richtige Verwendung des Webinterface. Grundsätzlich werden alle Textausgaben auf dem Bildschirm in englischer Sprache dargestellt. Diese Anleitung ist jedoch in deutscher Sprache verfasst, lediglich auf den Bildern, die zur Veranschaulichung dienen, ist der englische Text vorhanden.

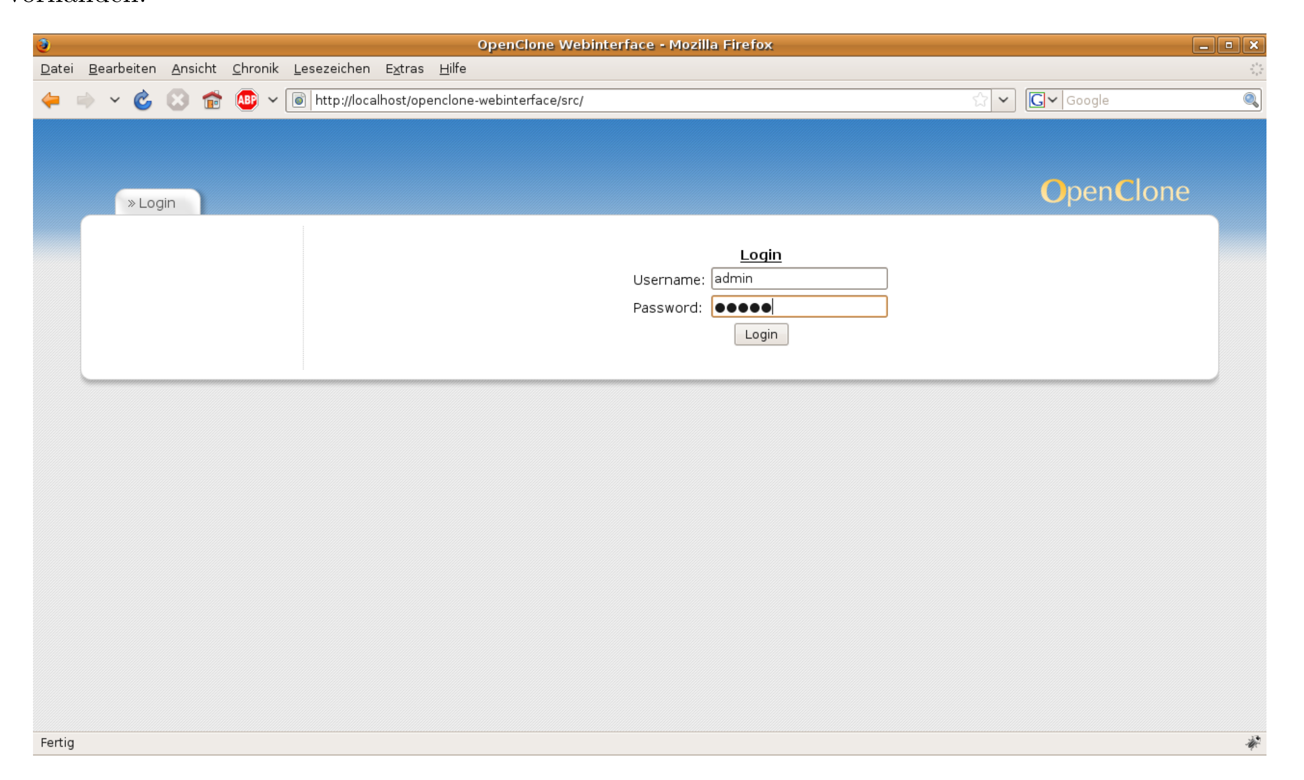


Abbildung B.1: Login Bildschirm

Allgemeines

Das Webinterface ist über einen Browser zu bedienen. Dabei muss die Adresse des Servers, auf dem das Webinterface installiert ist, in die Adressleiste eingegeben werden, um auf die Startseite zu gelangen. Auf der Startseite angelangt, muss man sich mit Benutzernamen und Passwort authentifizieren, um weitere Aktionen durchführen zu können. Diese Eingabemaske wird in Abbildung B.1 veranschaulicht.

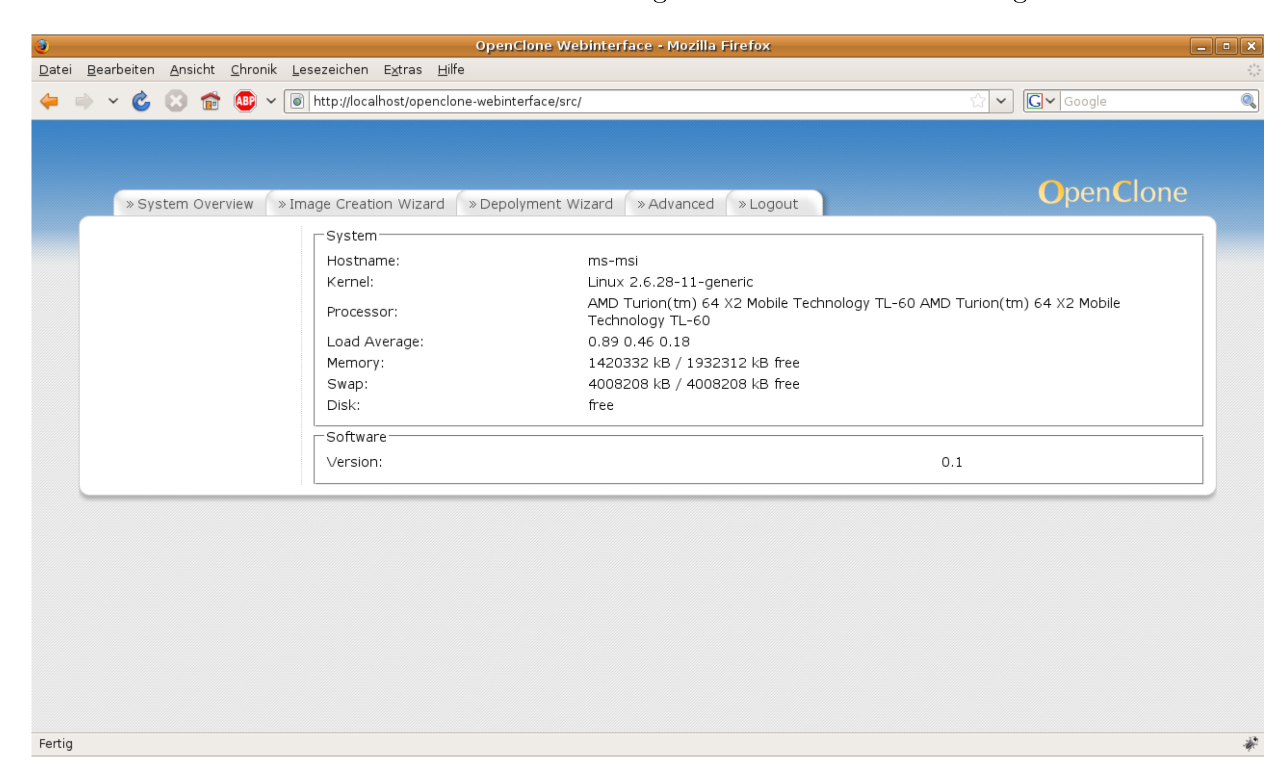


Abbildung B.2: System Overview (Übersichtsseite)

Nach erfolgreicher Authentifizierung ist eine Seite ersichtlich, welche Informationen über den Server, auf dem das Webinterface installiert ist, und die Versionsnummer der Software anzeigt. Diese Seite ist auch unter dem Menüpunkt *Overview* zu erreichen. Diese Seite wird in Abbildung B.2 dargestellt.

Nach allen getätigten Änderungen kann man sich im Menü unter dem Punkt *Logout* wieder abmelden. Dabei ist zu beachten, dass alle fertigen Tasks und deren zugehörigen Operationen aus der Datenbank gelöscht werden, um die Status- und Taskliste übersichtlich zu halten.

Anlegen eines Images

Um unkompliziert ein Festplattenabbild eines Clients erzeugen zu können, steht ein entsprechender Wizard unter dem Menüpunkt *Image Creation Wizard* zur Verfügung.

Als ersten Schritt muss der Computer ausgewählt werden, von dem ein Image angelegt werden soll. Der gewünschte Computer muss zu diesem Zeitpunkt die OpenClone Engine ausführen, damit er in der Liste der Rechner ersichtlich und auswählbar ist. Mit einem Klick auf *Next* gelangt man zum nächsten Schritt. Die Auswahl des Computers wird in Abbildung B.3 gezeigt.

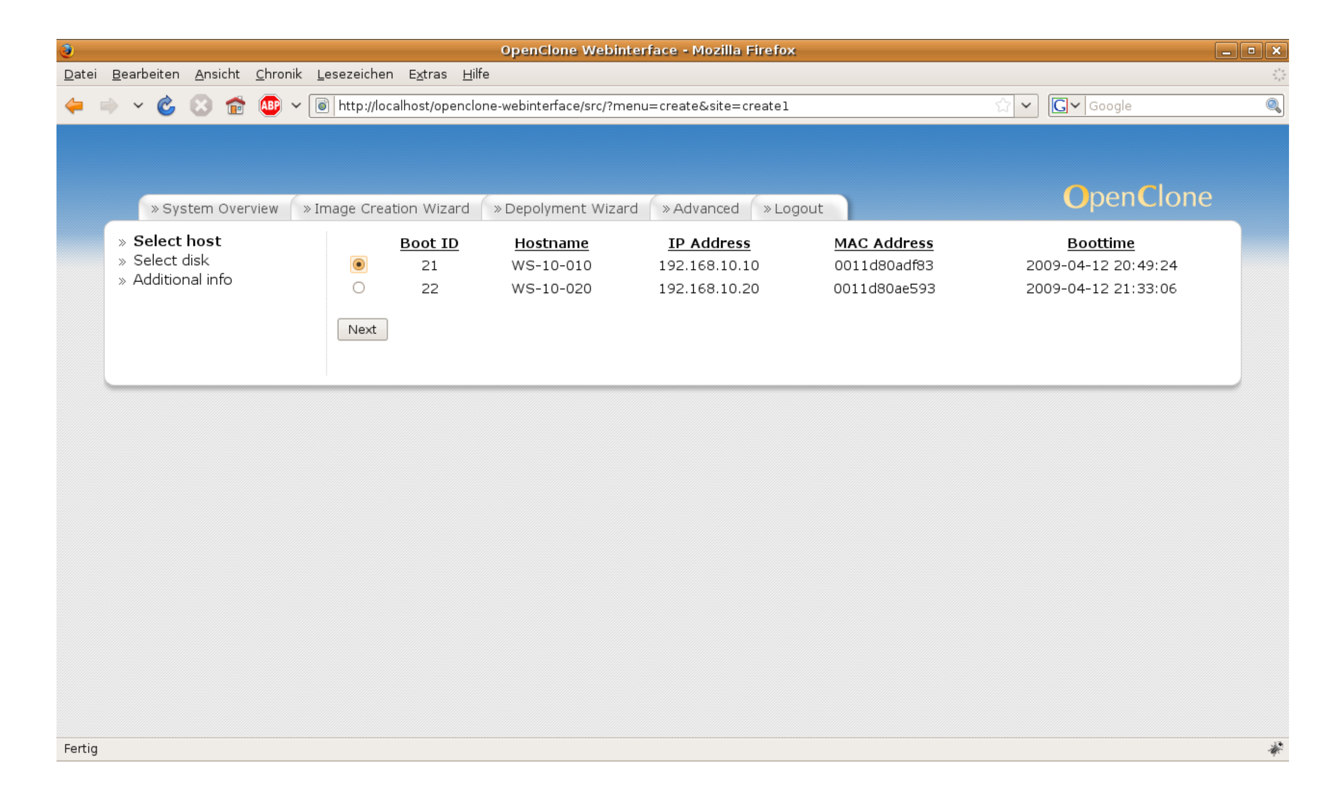


Abbildung B.3: Auswahl des Rechners im Image Creation Wizard

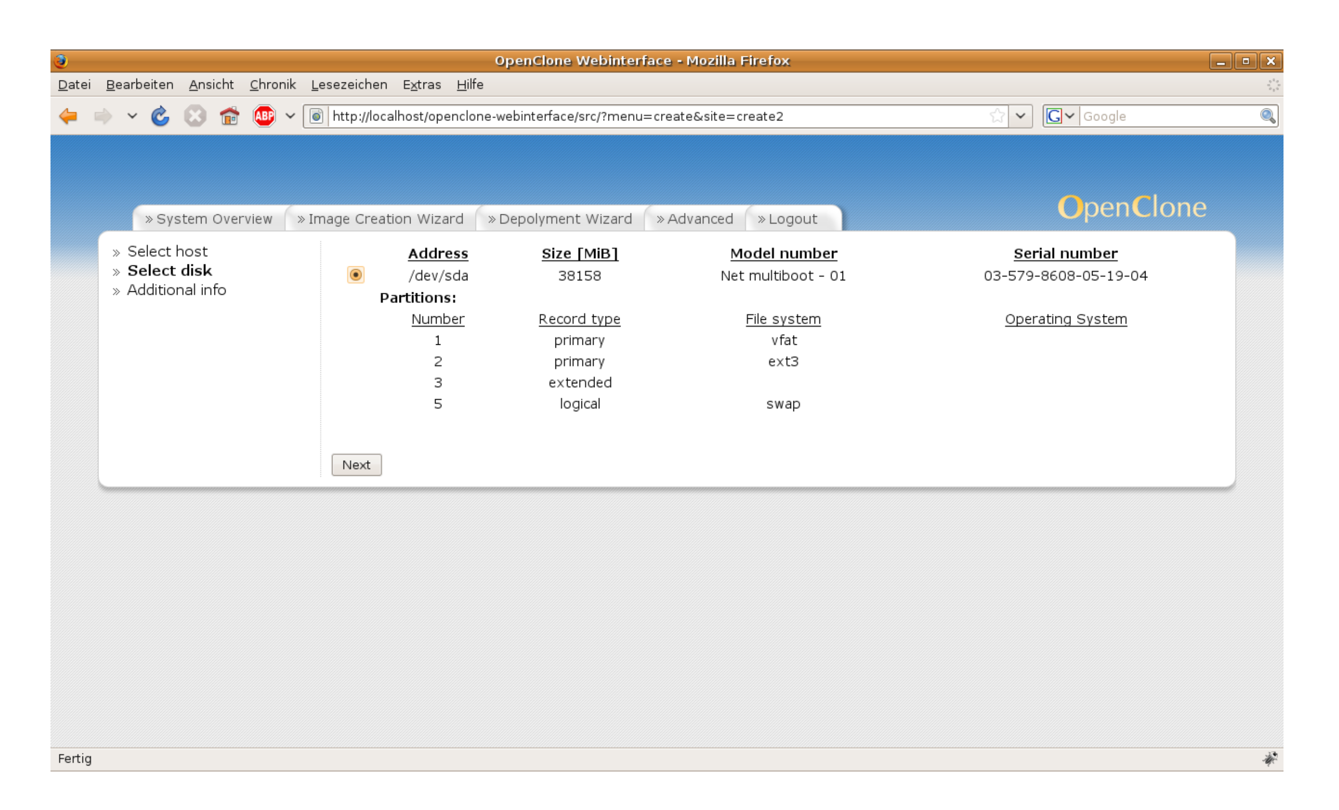


Abbildung B.4: Auswahl der Festplatte im Image Creation Wizard

Nach der Selektion eines Rechners muss im zweiten Schritt eine seiner Festplatten gewählt werden, von der das Abbild erstellt werden soll. Es werden alle Festplatten des Computers inklusive zugehöriger Partitionen angezeigt. Hat man eine Festplatte, gewählt (Abbildung B.4), gelangt man wieder mit

einem Klick auf *Next* zum letzten Schritt des Wizards.

Abbildung B.5: Zusätzliche Einstellungen im Image Creation Wizard

Als letzte Aktion wird eine Seite mit Informationen zum selektierten Rechner und der Festplatte angezeigt. Man muss nun noch einen Namen für das Image vergeben. Weiters kann man zwischen den beiden Kompressionsarten gzip und lzo wählen. Bei Verwendung von gzip kann eine kleinere Imagegröße erzielt werden, jedoch ist das Erstellen des Images mit lzo schneller erledigt. Bei Bedarf kann auch noch ausgewählt werden, ob der Rechner, von dem das Image erzeugt wird, nach dem Vorgang heruntergefahren oder neu gestartet werden soll. Die letzten Einstellungen für den Wizard sind in Abbildung B.5 dargestellt. Mit dem Klick auf *Finish* wird der Imagingprozess in die Datenbank eingetragen und kann vom Client abgearbeitet werden. Es wird nun ein Link sichtbar, mit dem man auf die Statusseite wechseln kann, wo man den Fortschritt des Prozesses verfolgen kann.

Am OpenClone Server wird für das Image ein Verzeichnis namens <Image ID> im Verzeichnis /data/ erzeugt, worin das Image gespeichert wird. Die Image ID wird für jedes Image in der Datenbank generiert.

Hinweis für Linux Betriebssysteme

Dieser Abschnitt beschreibt die Vorgehensweise beim Clonen von Betriebssystemen auf Basis von Linux, die in der Datei /etc/fstab den Eintrag des Root-Dateisystems mittels der Festplatten-ID standardmäßig eingestellt haben. Dieser Eintrag sieht folgendermaßen aus: `/dev/disk/by-id/<Festplatten-ID>-<Partition> /`. Die Festplatten-ID setzt sich aus Teilen der Modellbezeichnung und der Seriennummer jeder Festplatte individuell zusammen. Wird nun ein Image angelegt und auf andere Computer verteilt, kann auf diesen danach das Root-Dateisystem nicht gemountet werden, weil in diesen Rechnern

eine Festplatte mit eine anderen Seriennummer vorhanden ist. Somit ist auch die Festplatten-ID unterschiedlich und die Festplatte mit der in der Datei /etc/fstab angegebenen ID kann auf dem System nicht gefunden werden. Um dieses Problem zu umgehen müssen folgende Änderungen vor dem Erstellen des Images vorgenommen werden.

- In der Datei /etc/fstab muss jene Zeile geändert werden, welche das Mounten des Root-Dateisystems angibt, also die Zeile mit folgendem Aufbau: `/dev/disk/by-id/<Festplatten-ID> /`. In dieser Zeile muss die Festplatte anders angesprochen werden, geeignet hierfür ist die UUID[1], welche die ID des Dateisystems ist und auch nach dem Verteilen des Images gleich bleibt. Die Zeile sieht nach der Änderung folgendermaßen aus: `UUID=<UUID des Dateisystems> /` Weiters sollte diese Änderung in jeder Zeile gemacht werden, in der eine Partition mit der Festplatten-ID angegeben ist.
- In der Datei /boot/grub/menu.lst muss jene Zeile angepasst werden, die mit `kernel` beginnt. In dieser Zeile wird das Root-Dateisystem ebenfalls mit `root=/dev/disk/by-id/<Festplatten ID>` angegeben. Dies ist in `root=UUID=<UUID der Festplatte>` zu ändern, damit nach der Imageverteilung auf jedem Computer der Kernel auf der richtigen Festplatte gefunden und das System gestartet werden kann.

Es wurde ein Test mit SuSE 11.1 durchgeführt. Bei diesem Betriebssystem waren die genannten Änderungen nötig, um erfolgreich ein Image anlegen und wieder verteilen zu können.

Ein Image verteilen

Um ein bereits angelegtes Image auf einen oder mehrere Rechner zu verteilen, steht ebenfalls ein Wizard zur Verfügung. Er ist unter dem Menüpunkt *Deployment Wizard* erreichbar.

Im ersten Schritt muss das Image ausgewählt werden, welches verteilt werden soll. Auf der Auswahlseite werden alle verfügbaren Images inklusive Partitionen angezeigt. Diese Seite ist in Abbildung B.6 zu sehen. Durch einen Klick auf *Next* gelangt man zum nächsten Schritt.

Auf der nun erscheinenden Seite werden die Rechner gewählt, auf welche das Image verteilt werden soll. Dabei werden alle zur Zeit im OpenClone System eingeschalteten Hosts angezeigt. Man kann jedoch durch das Drop Down Menü ganz oben auf der Seite auswählen, welche Hosts zur Auswahl angezeigt werden sollen. Nach der Auswahl der Computer gelangt man mit *Next* zum letzten Schritt des Wizards. Die Auswahl der Hosts ist in Abbildung B.7 veranschaulicht.

Nach der Auswahl der Computer erfolgt nun der letzte Schritt des Wizards. Es muss die Festplatte gewählt werden, auf welche das Image gespeichert werden soll. Zur Auswahl stehen nur Festplatten, die in jedem gewählten Computer vorhanden sind und zusätzlich die selbe Geräteadresse haben. Die Größe der Festplatte in den einzelnen Hosts kann jedoch unterschiedlich sein. Ist in jedem Computer zum Beispiel unter der Adresse /dev/sda eine Festplatte vorhanden, wird sie zur Auswahl angezeigt. Ist in jedem Computer bis auf einem auch unter /dev/sdb eine Festplatte vorhanden, wird sie nicht

[1] Universally Unique Identifier

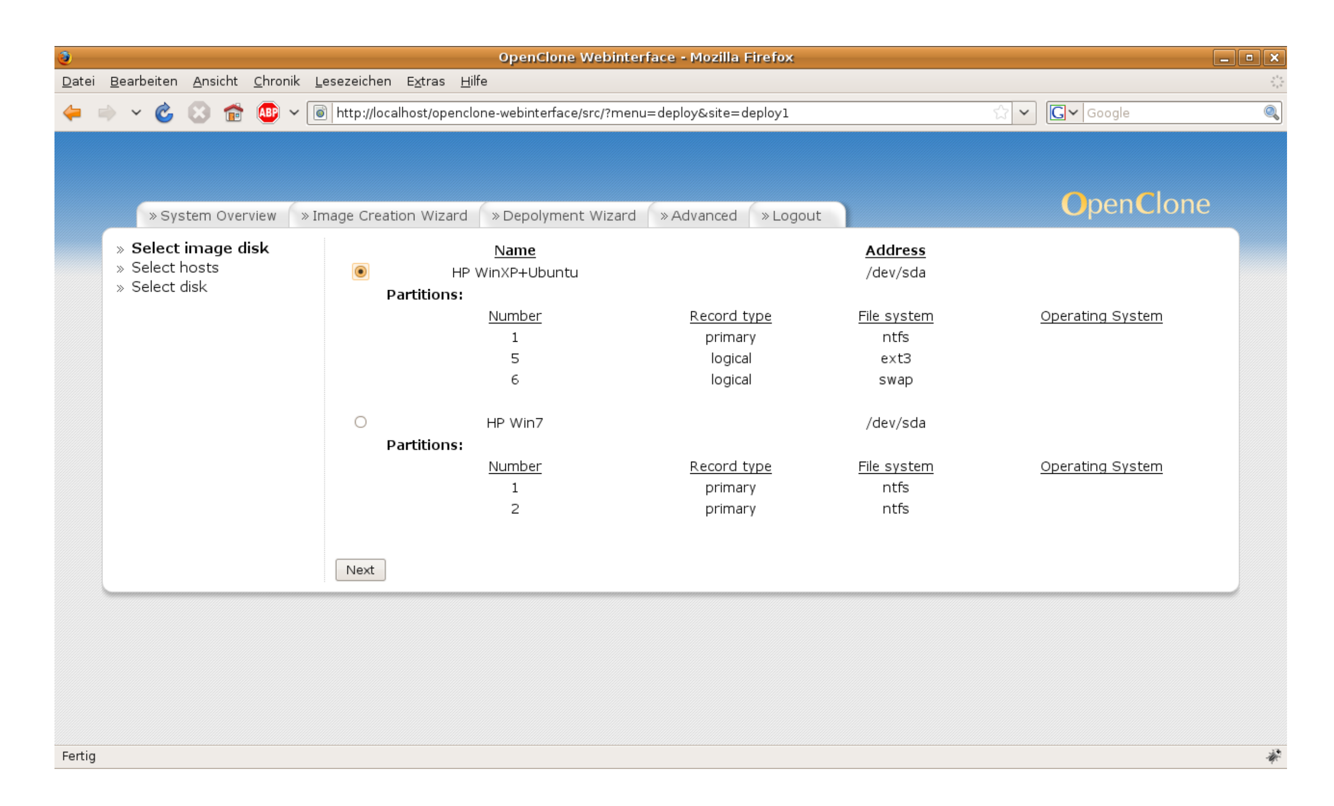


Abbildung B.6: Auswahl des Images im Deployment Wizard

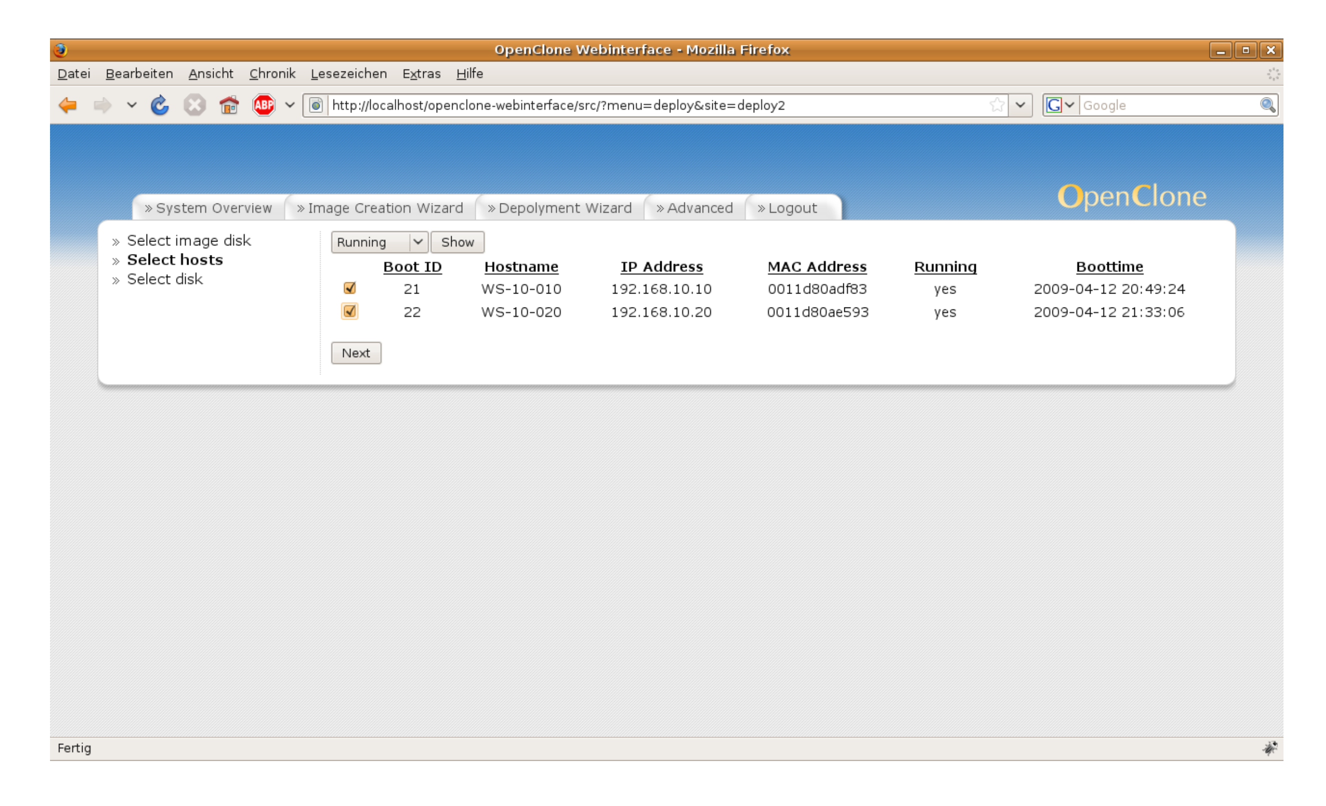


Abbildung B.7: Auswahl der Rechner im Deployment Wizard

angezeigt. Es ist zu beachten, dass im Webinterface auch die Größe der kleinsten Festplatte der Computer angezeigt wird. Ist auf dieser Festplatte zu wenig Platz, um das Image speichern zu können, kann diese Festplatte im Webinterface nicht ausgewählt werden.

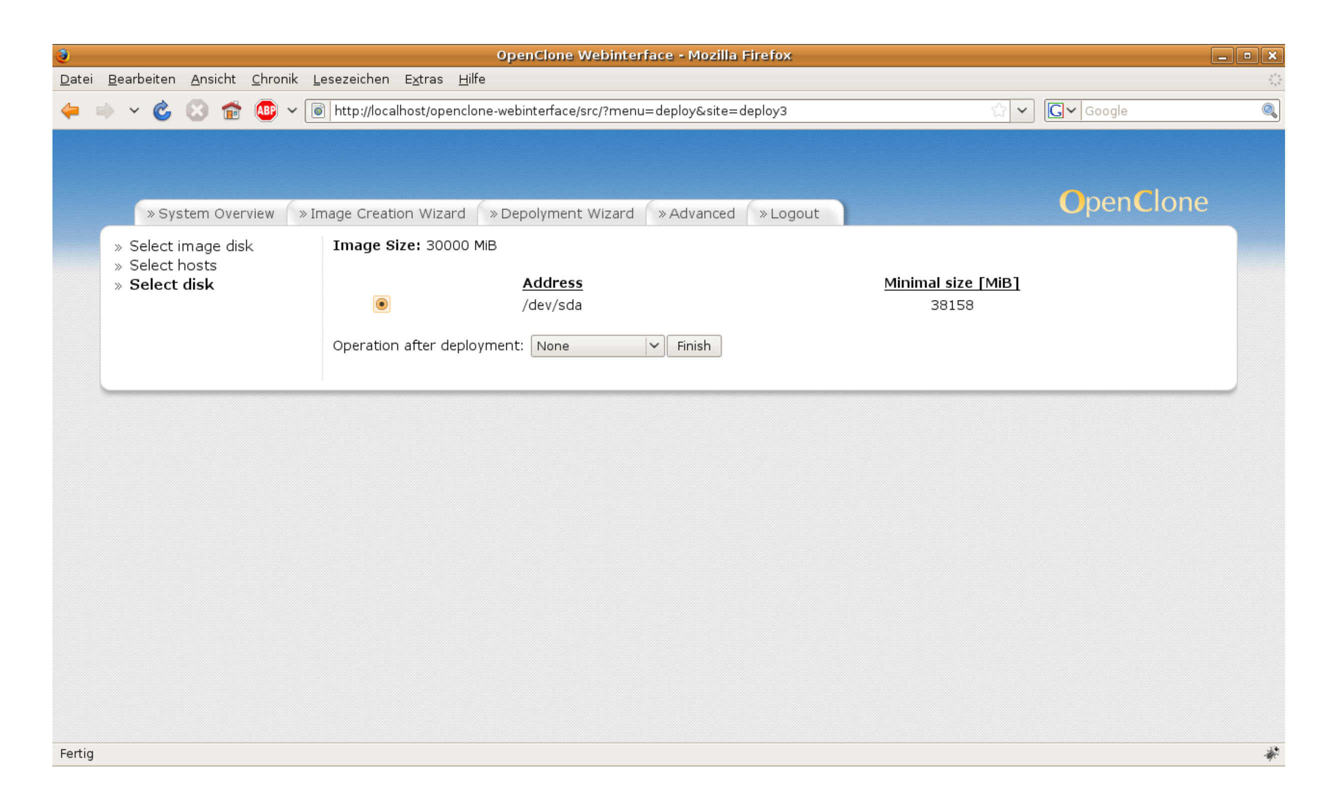


Abbildung B.8: Auswahl der Festplatte im Deployment Wizard

Im Normalfall sind in einer Lehrinstitution in einem Raum Rechner mit der selben Hardware installiert, somit ist auch die Festplatte jedes Rechners gleich groß und der Imageverteilung steht nichts im Weg. Nach der Auswahl der Festplatte, kann mit einem Klick auf *Finish* die Aufgabe in die Datenbank eingetragen und von den Clients abgearbeitet werden. Auch hier besteht die Möglichkeit, im Webinterface durch einen Link auf die Statusseite zu gelangen.

Das Image wird in jedem Computer auf die Festplatte gespeichert, die unter der Adresse zu erreichen ist, wie jene, die im Webinterface gewählt wird. Ist dies im Webinterface zum Beispiel /dev/sda, wird das Image auf jedem Client auf die Festplatte gespeichert, die unter /dev/sda erreichbar ist. Abbildung B.8 zeigt die Seite mit der Festplattenauswahl.

Erweiterte Möglichkeiten

Unter dem Menüpunkt *Advanced* kann man weitere Einstellungen und Informationen im OpenClone System ändern. Es erscheint ein weiteres Menü links im Bild, über welches man zu den einzelnen Einstellungsmöglichkeiten gelangt.

Hostliste

Der Punkt *Hostlist* führt zu einer Auflistung aktuell eingeschalteter Hosts in OpenClone. Die angezeigten Hosts können durch das Drop Down Menü am Beginn der Seite gezielt ausgewählt werden. Durch Auswahl einer Gruppe werden nur die zu dieser Gruppe gehörigen Hosts angezeigt. Weiters besteht die Möglichkeit, Hosts zu selektieren und folgende Operationen auf diese auszuführen.

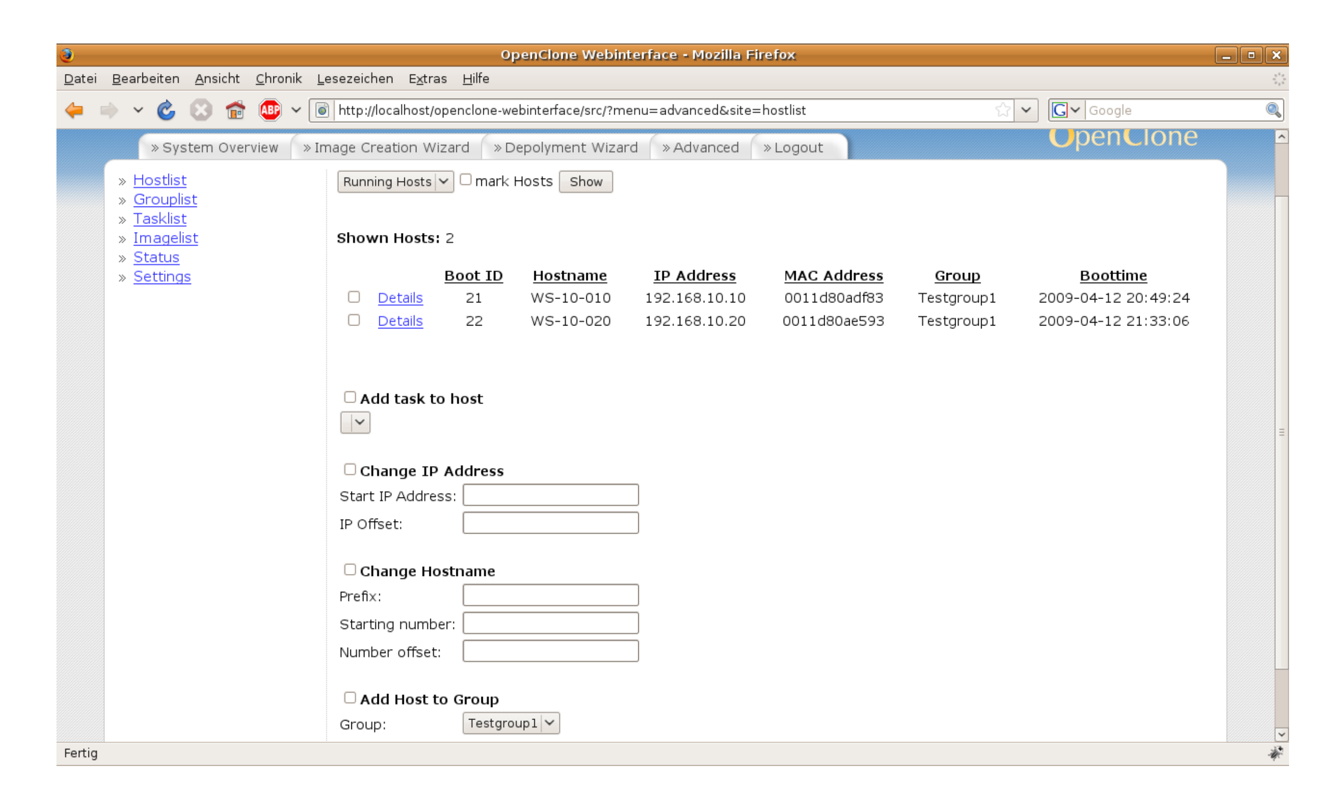


Abbildung B.9: Hostliste im Menü Advanced

- Ein Task kann den gewählten Hosts zugewiesen werden, damit er von diesen ausgeführt wird
- Die IP Adresse kann geändert werden. Dabei bekommt der erste gewählte Host die angegebene IP Adresse und alle folgenden Hosts eine IP Adresse im selben Netz, jedoch im letzten Oktett der Adresse eine immer um den Wert, der bei Offset angegeben wurde, erhöhte Nummer
- Die Hostnamen der Computer kann geändert werden. Dabei ist zu beachten, dass der Prefix bei jedem Host am Beginn des Hostnamens steht und der Postfix für jeden Host eine dreistellige Nummer ist. Dabei wird der Abstand zwischen den Nummern durch den Offset angegeben
- Die Rechner können einer Gruppe zugewiesen werden

Durch Auswahl mehrerer Checkboxen können die entsprechenden Aktionen gleichzeitig ausgeführt werden. Eine Übersicht dieser Seite ist in Abbildung B.9 gezeigt.

Durch den Link *Details* bei jedem Host wird eine Seite wie in Abbildung B.10 gezeigt, auf welcher weitere Informationen zu dem gewählten Host dargestellt sind. Weiters lassen sich auf dieser Seite Informationen des Hosts wie Seriennummer, IP Adresse, MAC Adresse und Hostname ändern.

Gruppenliste

Bei Klick auf den Link *Grouplist* erscheinen alle vorhanden Gruppen. Durch Eingabe eines Gruppennamens und Klick auf *Add Group* kann eine weitere Gruppe angelegt werden. Weiters kann eine Gruppe gewählt und durch Auswahl von *delete* gelöscht werden. Falls dieser Gruppe Hosts zugeordnet waren, werden diese danach keiner Gruppe zugeordnet. Diese Seite wird in Abbildung B.11 dargestellt.

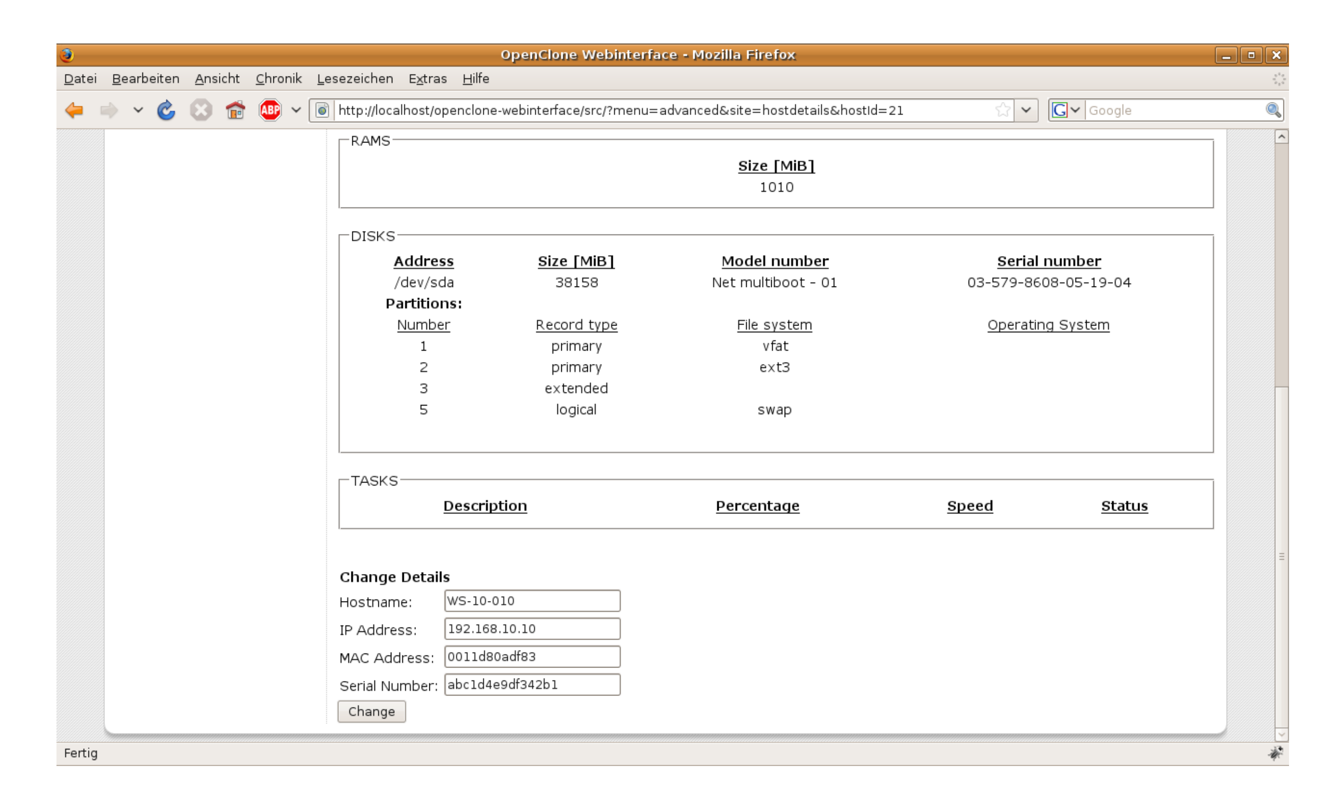


Abbildung B.10: Details zu einem Host

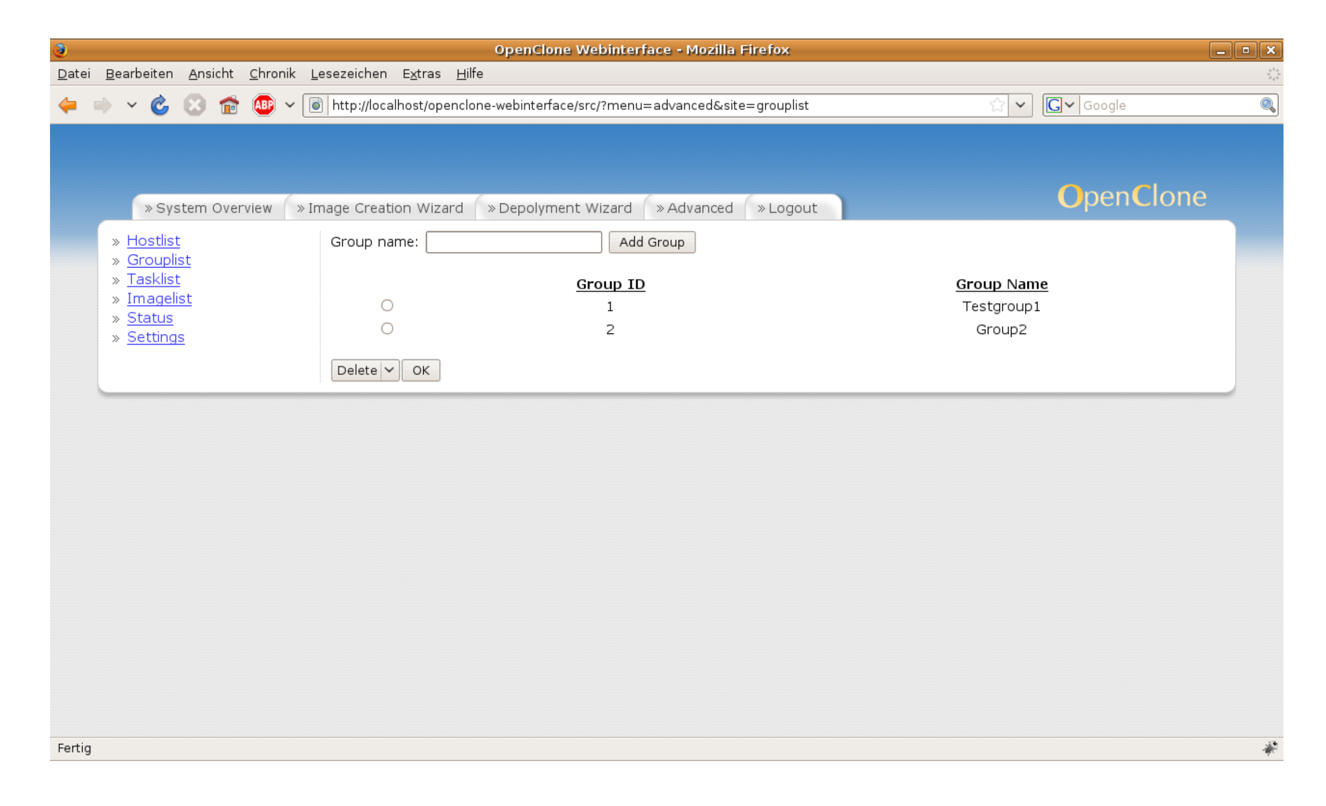


Abbildung B.11: Gruppenliste im Menü Advanced

Durch Auswahl von *edit* im Drop Down Menü gelangt man auf eine Seite, auf welcher Hosts aus der gewählten Gruppe gelöscht werden können. Dies ist in Abbildung B.12 veranschaulicht.

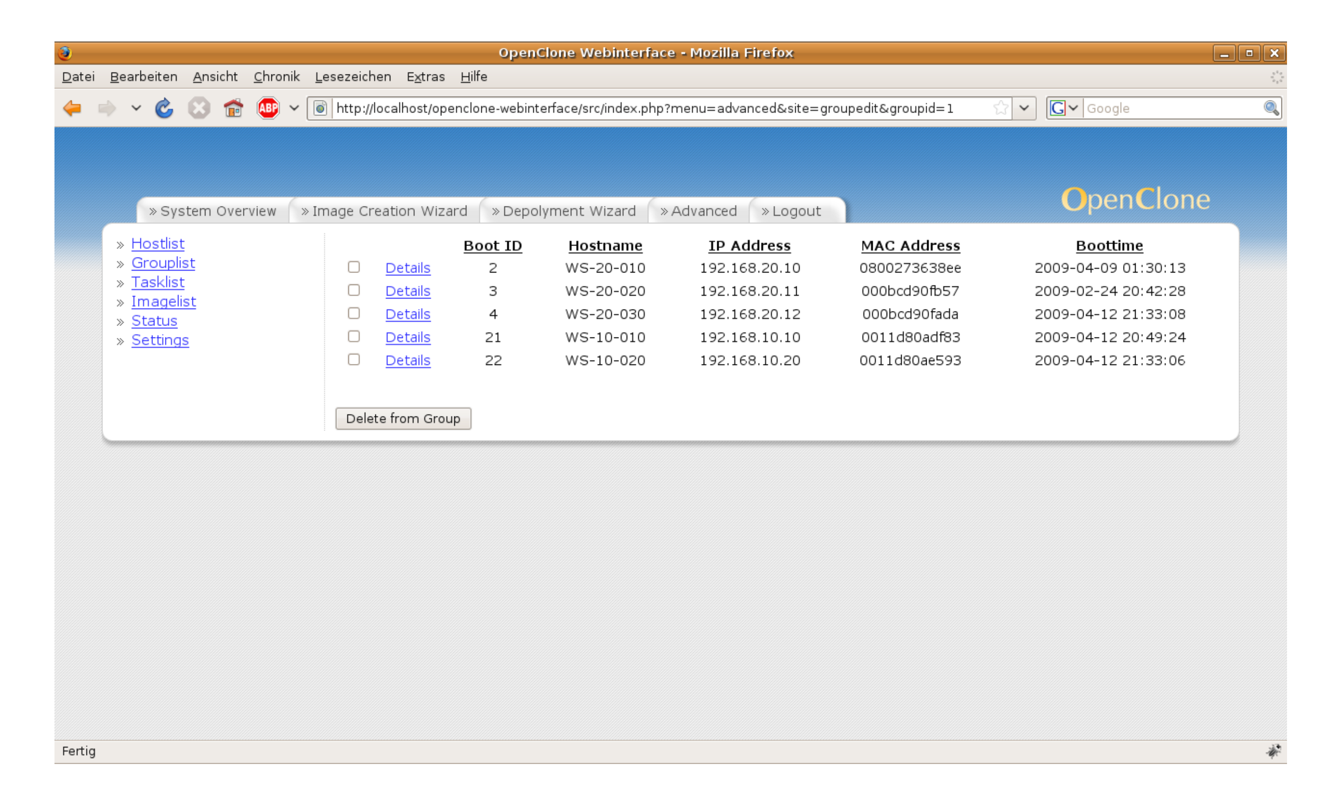


Abbildung B.12: Seite zum Editieren einer Gruppe

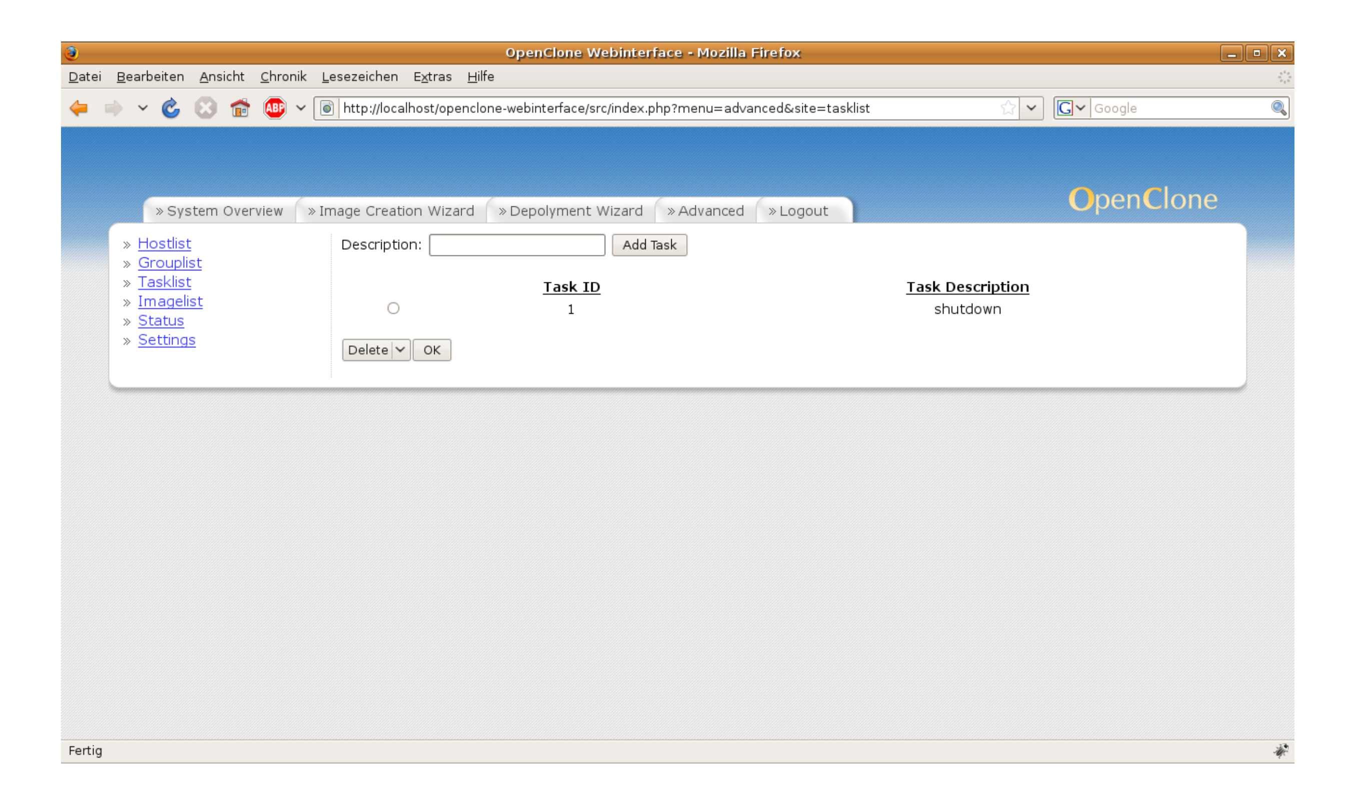


Abbildung B.13: Taskliste im Menü Advanced

Taskliste

Die Seite, welche bei Klick auf *Tasklist* erscheint, listet alle im System eingetragenen Tasks auf. Man kann auf dieser Seite nach Eingabe eines Tasknamen und Klick auf *Add Task* einen neuen Task anlegen.

Weiters kann ein Task ausgewählt werden und durch Auswahl von *delete* im Drop Down Menü gelöscht werden. Die Tasklistseite ist in Abbildung B.13 zu sehen.

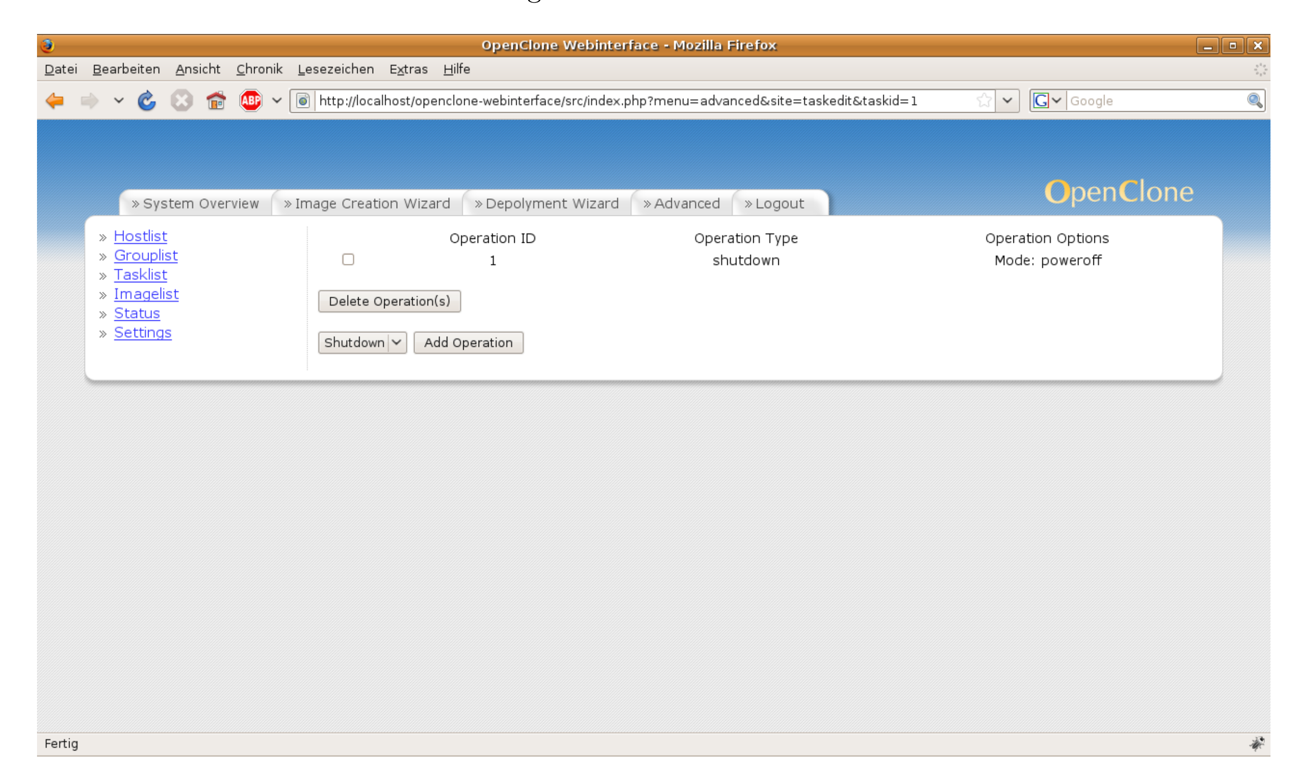


Abbildung B.14: Seite zum Editieren eines Task

Wenn man im Drop Down Menü edit wählt, gelangt man auf eine Editierseite, auf der man dem Task zugewiesene Operationen löschen oder dem Task eine neue Operation hinzufügen kann. Dabei stehen im entsprechenden Drop Down Menü *Shutdown* und *Reboot* zur Verfügung. *Shutdown* legt eine Operation zum Herunterfahren der Clients an, *Reboot* eine zum Neustart. Diese Einstellungen sind in Abbildung B.14 ersichtlich.

Imageliste

Diese Seite ist unter dem Menüpunkt *Imagelist* zu sehen und zeigt alle verfügbaren Images inklusive deren Größe an. Hier besteht die Möglichkeit, ein einzelnes oder mehrere Images zu löschen. Alle mit dem Image in Verbindung stehenden Tasks und deren Operationen werden dabei ebenfalls aus der Datenbank gelöscht. Weiters kann der Name eines Images geändert werden. Dies ist jedoch nur möglich, wenn genau ein Image ausgewählt ist. Die Auflistung der Images ist in Abbildung B.15 veranschaulicht.

Generelle Einstellungen

Der Menüpunkt *Settings* führt zu einer Seite mit generellen Einstellungsmöglichkeiten. In der Version 0.1.0 des OpenClone Webinterface besteht hier lediglich die Möglichkeit, Benutzernamen und Passwort des aktuell eingeloggten Benutzers zu ändern. Der Name kann jedoch nur geändert werden, falls noch keine anderer Benutzer mit gewünschtem Namen in der Datenbank existiert. Zur Vergabe eines neuen

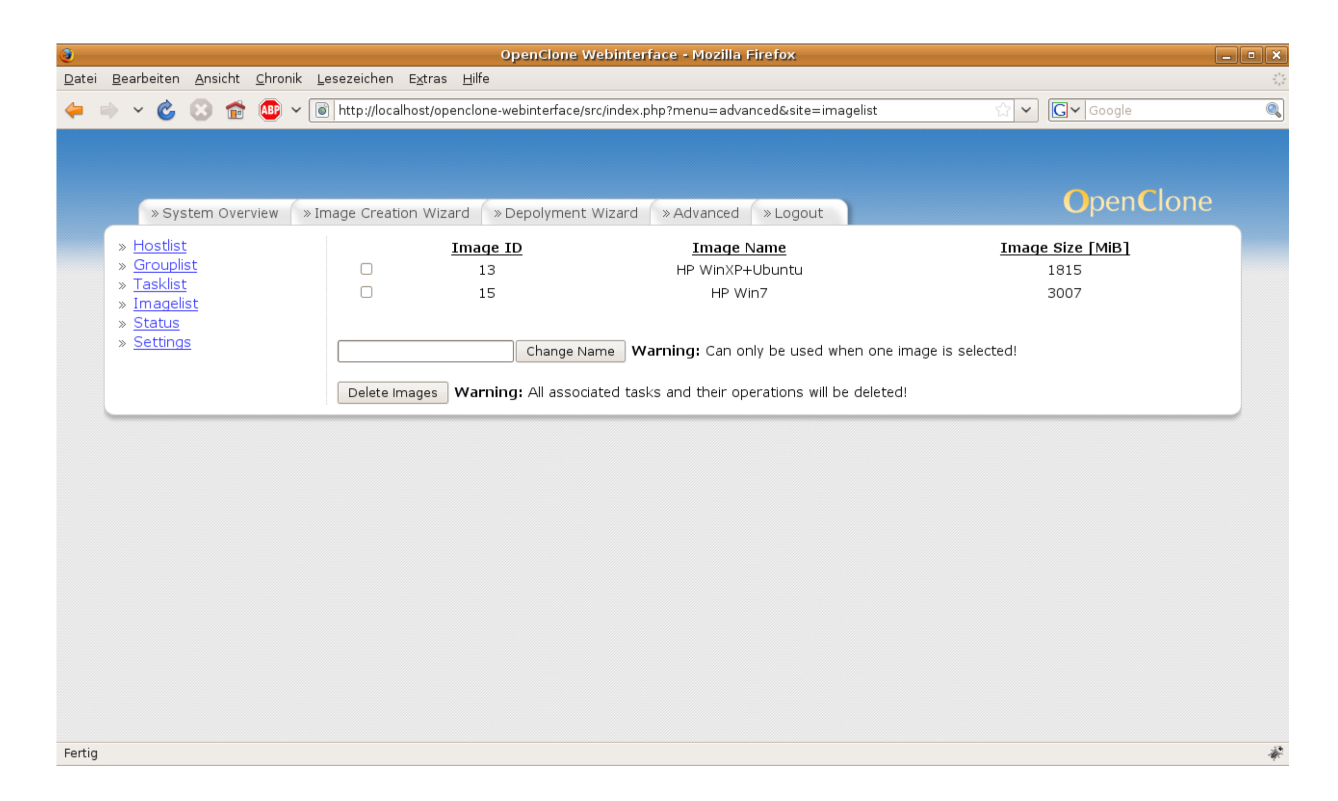


Abbildung B.15: Imageliste

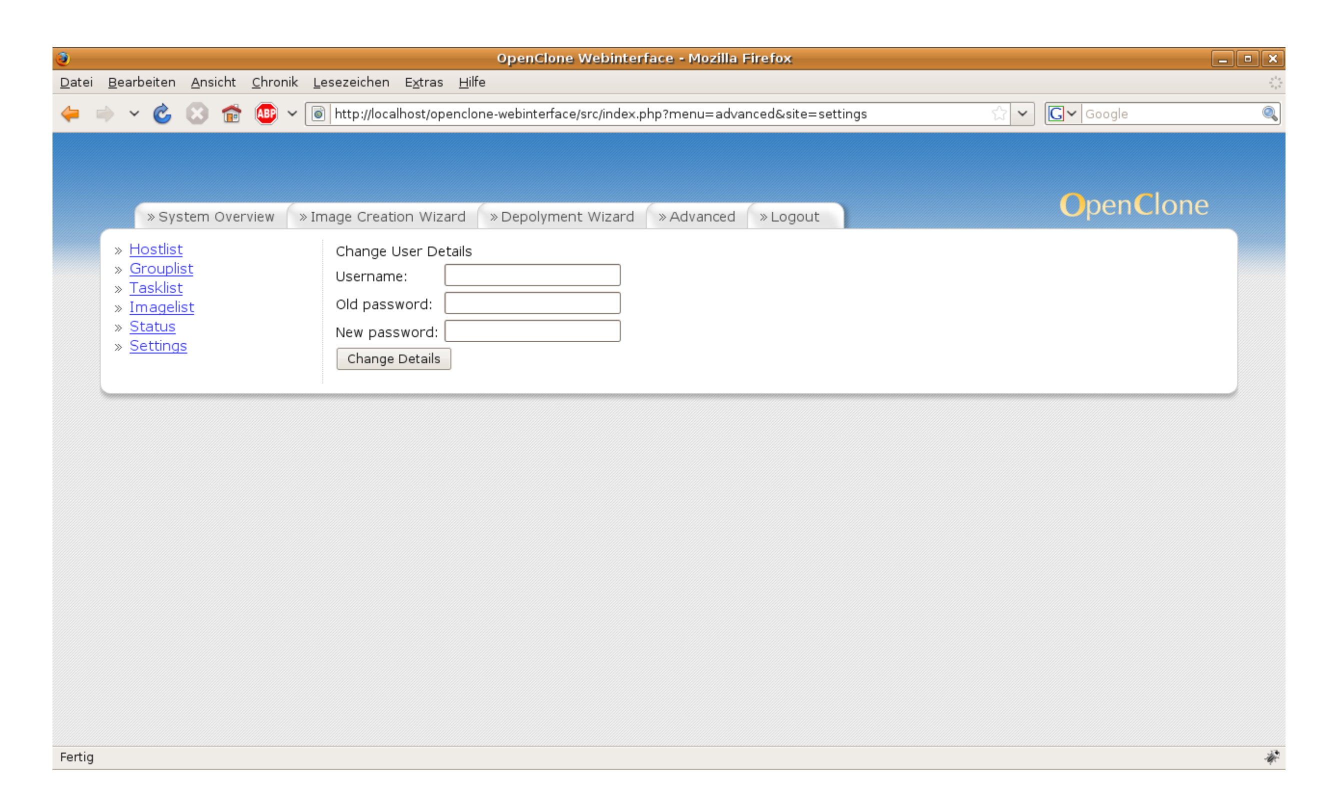


Abbildung B.16: Seite zum Ändern der Benutzerinformationen

Passwortes muss aus Sicherheitsgründen das alte Passwort ebenfalls eingegeben werden. Das Ändern der Benutzerinformationen ist in Abbildung B.16 dargestellt.

B.3 Erweiterte Konfiguration

In diesem Abschnitt werden fortgeschrittene Konfigurationsmöglichkeiten erklärt, welche aber eine manuelle Einrichtung erfordern.

B.3.1 Subnetze hinzufügen

OpenClone bietet bei manueller Konfiguration die Möglichkeit, mit Clients in mehreren Subnetzen zu arbeiten.

Dazu ist es notwendig einerseits auf dem Server eine IP-Adresse in jedem gewünschten Subnetz einzurichten und andererseits die `pxelinux`-Konfiguration für dieses Subnetz anzupassen.

Netzwerkkonfiguration unter Ubuntu

Die Netzwerkkonfiguration von Ubuntu ist in der Datei `/etc/network/interfaces` enthalten. Zum Hinzufügen einer weiteren IP-Adresse muss mit den Zeilen aus Listing B.1 ein neues Subinterface in der Konfiguration hinzugefügt werden. Die Netzwerkschnittstelle eth0 muss eventuell angepasst werden und die Subinterface Nummer, welche mit Doppelpunkt vom Interface getrennt ist, für jedes Subinterface immer weiter erhöht werden. Mit der IP-Adresse und Netmask muss eine IP-Adresse im gewünschten Subnetz konfiguriert werden.

```
1 auto eth0:1
2 iface eth0:1 inet static
3 address 192.168.13.206
4 netmask 255.255.255.0
```

Listing B.1: Ubuntu Netzwerkkonfiguration /etc/network/interfaces

Anschließend ist zum Anwenden der Konfiguration der Befehl `sudo /etc/init.d/networking restart` oder ein Neustart notwendig.

PXELINUX-Konfiguration

Damit die Clientcomputer beim Booten über PXE die richtigen Parameter für das Root Dateisystem und die OpenClone Engine erhalten, muss für jedes Subnetz eine eigene `pxelinux`-Konfiguration vorhanden sein. Die Konfigurationen sind im TFTP Verzeichnis, also normalerweise unter `/tftpboot`, im Verzeichnis `pxelinux.cfg` enthalten. Die Konfigurationen können, wie in Kapitel 4.6.2 beschrieben, unterschiedliche Namen haben. Einerseits die MAC-Adresse der Client Netzwerkkarte, die IP-Adresse, sowie gesamte Subnetze oder die Standardkonfiguration `default`. Am Zielführendsten ist die Verwendung von Subnetzen. Dazu muss die Netzadresse in die hexadezimale Schreibweise umgewandelt werden und anschließend je nach Netzmaske eine bestimmte Anzahl an Ziffern vom Ende entfernt werden. Die Umwandlung kann, wie in Listing B.2 gezeigt, mit Hilfe des `gethostip` Programms erfolgen.

```
1 $ gethostip 192.168.13.0
2 192.168.13.0 192.168.13.0 C0A80D00
```

Listing B.2: Umwandlung der Netzadresse in Hex

Anschließend werden für eine 24 Bit Netzmaske die letzten 2 Ziffern abgeschnitten und als Dateiname für die Konfiguration verwendet, also in Fall von 192.168.13.0/24 wird der Dateiname C0A80D verwendet. Auf die Großschreibung ist zu achten. Im Listing B.3 wird der Inhalt der neuen Konfigurationsdatei für den Server mit der IP 192.168.13.206 gezeigt. Am Zielführendsten wird das Kopieren und anschließende Anpassen der Standardkonfiguration mit dem Namen `default` sein.

```
1 DEFAULT openclone
2 PROMPT 0
3 MENU TITLE OpenClone BootMenu
4 LABEL openclone
5   KERNEL ubuntu/vmlinuz
6   APPEND initrd=ubuntu/initrd.img root=/dev/nfs nfsroot=192.168.13.206:/
      nfsroot ip=dhcp ocws=http://192.168.13.206:18080/OpenCloneEngineService
```

Listing B.3: PXELINUX Konfiguration `C0A80D` für das Subnetz 192.168.13.0

B.3.2 SSL-Unterstützung für das Webinterface

Um eine verschlüsselte Übertragung zwischen dem Webbrowser eines Verwaltungsclients und dem Apache Webserverdienst auf dem Server, auf welchem das Webinterface vorhanden ist, zu ermöglichen, muss die SSL Unterstützung für Apache eingerichtet sein. Unter Ubuntu Linux 8.04 Hardy Heron sind folgende Schritte notwendig, um nicht nur über den HTTP Port 80, sondern auch über den HTTPS Port 443 mit dem Webserver kommunizieren zu können.

1. Im ersten Schritt muss ein Zertifikat mit dem Befehl `sudo openssl req -new -x509 -days 365 -nodes -out /etc/apache2/ssl/apache.pem -keyout /etc/apache2/ssl/apache.pem` erstellt werden. Dabei gibt der Wert hinter dem Argument `-days` die Gültigkeitsdauer des Zertifikates an.

2. Der Apache Webserver muss so konfiguriert werden, dass er auch auf Port 443 für Anfragen zur Verfügung steht. Um dies zu ermöglichen wird die Datei /etc/apache2/ports.conf um folgende Zeilen ergänzt (Listing B.4).

```
1 <IfModule mod_ssl.c>
2     Listen 443
3 </IfModule>
```

Listing B.4: Apache Portkonfiguration

3. Der Apache Webserver muss mit dem Befehl `/etc/init.d/apache2 reload` die Konfiguration neu laden.

4. Das SSL Modul wird mittels `sudo a2enmod ssl` aktiviert.

5. Für die Seite mit aktiviertem SSL wird ein virtueller Host angelegt. Dazu muss die Konfigurationsdatei /etc/apache2/sites-available/ssl mit dem Inhalt, der in Listing B.5 dargestellt ist, angelegt werden. Es werden hierbei mit Befehlen das SSL für diesen Host aktiviert und die Zertifikatsdatei angegeben.

```
1 NameVirtualHost *:443
2 <virtualhost *:443>
3 ...
4         SSLEngine On #SSL aktivieren
5         SSLCertificateFile /etc/apache2/ssl/apache.pem #Zertifikatsdatei
              angeben
6 ...
7 </virtualhost>
```

Listing B.5: Apache Konfiguration des virtuellen Hosts mit SSL

6. Die Konfiguration des virtuellen Hosts muss mit `sudo a2ensite ssl` aktiviert werden.

7. Zum Schluss wird der Webserver durch den Befehl `/etc/init.d/apache2 restart` neu gestartet. Danach ist auch eine verschlüsselte Kommunikation möglich.

B.3.3 Multicast-Unterstützung von Switches aktivieren

Um die Multicast-Übertragung im Ethernet Netzwerk effizienter zu machen, muss bei den Switches das IGMP Snooping aktiviert werden. Dadurch analysiert der Switch alle eintreffenden IGMP Nachrichten, kann die Zugehörigkeit zu bestimmten Multicast Gruppen ermitteln und Multicast Übertragungen nur an die Rechner einer Multicast Gruppe weiterleiten, anstatt sie wie Broadcasts an alle auszusenden. (siehe Kapitel 4.3.5)

HP ProCurve Switches 1600M, 2424M, 4000M und 8000M[2]

Zum Aktivieren der IGMP Unterstützung über die Switch Console bzw. über Telnet sind folgende Schritte ausgehend vom Hauptmenü notwendig:

1. *Switch Configuration* auswählen
2. *Advanced Features* auswählen
3. *IP Multicast (IGMP) Service* auswählen
4. *IGMP Enable* auf *Yes* setzen
5. *Forward with High Priority* am Besten auf *No* belassen
6. Am besten alle Ports auf *Auto* belassen
7. Anschließend die Konfiguration mit *Save* speichern

[2] [Com09a, S. 6-95ff]

HP ProCurve Switches 2512 und 2524[3]

Listing B.6 zeigt die Konfiguration der IGMP Unterstützung für das VLAN 10 auf der Kommandozeile über die Switch Console oder Telnet.

```
HP_2524# configure
HP_2524(config)# vlan 10 ip igmp
HP_2524(config)# exit
HP_2524# show ip igmp config

IGMP Service

VLAN ID VLAN NAME    IGMP Enabled Forward with High Priority Querier Allowed
------- ------------ ------------ -------------------------- ---------------
1       DEFAULT_VLAN No           No                         Yes
10      VLAN10       Yes          No                         Yes
12      VLAN12       No           No                         Yes
```

Listing B.6: IGMP Unterstützung für VLAN 10 aktivieren

[3] [Com09b, S. 9-91ff]

	**HÖHERE TECHNISCHE LEHRANSTALT für Informationstechnologie** **YBBS AN DER DONAU**
	Ausbildungsschwerpunkt: **Netzwerktechnik**

DIPLOMARBEIT
DOKUMENTATION

Namen der Verfasser/innen	GNEDT David STEINER Manuel
Jahrgang / Klasse Schuljahr	5AHITN 2008/09
Thema der Diplomarbeit	Automatisierung von Installationsroutinen in Netzwerken
Kooperationspartner	-

Aufgabenstellung	Ziel der Diplomarbeit soll der Vergleich von einigen bereits existierenden Cloning Systemen sein, deren Stärken und Schwächen aufzeigen und anschließend, teilweise aufbauend auf diesen Produkten, eine Entwicklung eines automatisierten Systems zum Installieren und Konfigurieren vollständiger Systeme vorzunehmen. Bei dem zu entwickelnden Produkt muss ein möglichst autarkes System implementiert werden, sodass der Administrator von vielen zeitaufwändigen Installationsarbeiten entkoppelt werden kann. Hauptaugenmerk wird auf dem Einsatz im Schulzentrum Ybbs liegen, wobei im Rahmen der GPL das Produkt auch anderen Schulen und Institutionen zugänglich gemacht werden kann.

Realisierung	Die Entwicklung des eigenen Cloningsystems wurde basierend auf den aus der Analyse von existierenden Cloninglösungen gewonnnen Informationen durchgeführt. Bei der Entwicklung wurde besonders auf die Bedürfnisse des Schulzentrums eingegangen, um die Software für den dortigen Gebrauch zu optimieren. Weiters wurde auf einfache Handhabungsmöglichkeit und Modularität geachtet. Somit ist eine Substitution einzelner Teilkomponenten oder das Einpflegen weiterer Softwareteile ohne größere Umstände realisierbar.

Ergebnisse	Ergebnis ist das Cloningsystem OpenClone, welches alle Pflichtanforderungen und einige Wunschanforderungen erfolgreich implementiert. Das System konnte im Schulzentrum Ybbs a.d. Donau getestet werden und wird dort auch für den praktischen Einsatz installiert. Durch die Verwendung einer freien Softwarelizenz und der Veröffentlichung als Open Source Projekt unter der GNU General Public License wird die entwickelte Lösung auch anderen Schulen und Institutionen zur Verfügung gestellt.

Typische Grafik, Foto etc. (mit Erläuterung)

OpenClone Client
OpenClone Client
OpenClone Server
Verwaltungsclient
OpenClone Client
...

Die Grafik zeigt den grundsätzlichen Aufbau von OpenClone. Der Verwaltungsclient greift auf das Webinterfaces des OpenClone Servers zu, welcher die OpenClone Clients steuert.

OpenClone Webinterface - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://localhost/openclone-webinterface/src/?menu=create&site=create3
Google
OpenClone
» System Overview » Image Creation Wizard » Depolyment Wizard » Advanced » Logout
» Select host
» Select disk
» **Additional info**

HOST

Boot ID	Hostname	IP Address	MAC Address	Boottime
21	WS-10-010	192.168.10.10	0011d80adf83	2009-04-12 20:49:24

DISK

Address	Size [MiB]	Model number	Serial number
/dev/sda	38158	Net multiboot - 01	03-579-8608-05-19-04

Partitions:

Number	Record type	File system	Operating System
1	primary	vfat	
2	primary	ext3	
3	extended		
5	logical	swap	

Image name: IMAGE1
Image compression: gzip
Operation after creation: None
Finish
Fertig

Die Grafik zeigt das OpenClone Webinterface, welches zur Steuerung des Cloningsystems dient. Dargestellt ist der letzte Schritt des Asisstenten zur Erstellung eines Festplattenabbildes eines Rechners.

Teilnahme an Wettbewerben, Auszeichnungen	-

Möglichkeiten der Einsichtnahme in die Arbeit	IT-HTL Ybbs a. d. Donau

Approbation (Datum / Unterschrift)	Prüfer/in	Abteilungsvorstand / Direktor/in

	COLLEGE of ENGINEERING for INFORMATION TECHNOLOGY YBBS AN DER DONAU
	Educational focus: **Network Technology**

DIPLOMA THESIS
Documentation

Author(s)	GNEDT David STEINER Manuel
Form Academic year	5AHITN 2008/09
Topic	Automation of installation routines in networks
Co-operation partners	-

Assignment of tasks	Goal of the diploma thesis is the comparison of already existing cloning systems, showing the pros and cons of them. Furthermore an automated system for installation and configuration of systems should be developed, based on informations gathered from the comparison. For freeing the administrator from lots of time-consuming installation works the system should be implemented autarkic as much as possible. The focus should be on the use in the Schulzentrum Ybbs, whereas the product can also be used by other schools, because of licensing the product under the open source license GPL.

Realization	The development of the own cloning system was based on the informations gathered from the analysis of existing cloning solutions. While the implementation, we focused on the requirements of the Schulzentrum for optimizing the software usability. Moreover an eye was kept on the modularity, so a substitution of the software components or adding new components is easily possible.

Results	The result of the diploma thesis is the cloning system OpenClone, which successfully implements all essential requirements and some desired requirements. The system was tested in the Schulzentrum Ybbs, where it also will be installed for practical use. Due to the usage of a free software license and publishing as open source project under the GNU General Public License the developed system can be used by other schools or institutions.

Illustrative graph, photo (incl. explanation)	This figure shows the general structure of OpenClone. The management client connects to the webinterface of the OpenClone server, which controls the OpenClone clients. This figure shows the OpenClone webinterface, which is used for controlling the cloning system. Shown is the last step of the Image Creation Wizard, which is used for creating a hard disk image of a computer.

Participation in competitions Awards	-

Accessibility of diploma thesis	College of Engineering for Information technology Ybbs a. d. Donau

Approval (Date / Sign)	Examiner	Head of Department / College